THE MODULAR CLOCK ALGORITHM FOR BLIND
RENDEZVOUS

THESIS

Nicholas C. Theis, Captain, USAF

AFIT/GCS/ENG/09-08

AFIT/GCS/ENG/09-08

The Modular Clock Algorithm for Blind Rendezvous

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Nicholas C. Theis

Captain, USAF

March 2009

AFIT/GCS/ENG/09-08

The Modular Clock Algorithm for Blind Rendezvous

Nicholas C. Theis
Captain, USAF

Approved:

/signed/	9 March 2009
Capt R.W. Thomas, PhD (Chairman)	Date

/signed/	9 March 2009
Dr. K.M. Hopkinson (Member)	Date

/signed/	9 March 2009
Dr. R.K. Martin (Member)	Date

AFIT/GCS/ENG/09-08

Abstract

This thesis examines the problem in initializing communications whereby cognitive radios need to find common spectrum with other cognitive radios, a process known as frequency rendezvous. It examines the rendezvous problem as it exists in a dynamic spectrum access cognitive network. Specifically, it addresses the problem of rendezvous in an infrastructureless environment. A new algorithm, the modular clock algorithm, is developed and analyzed as a solution for the simple rendezvous environment model, coupled with a modified version for environment models with less information. The thesis includes a taxonomy of commonly used environment models, and analysis of previous efforts to solve the rendezvous problem. Mathematical models and solutions used in applied statistics are analyzed for use in cognitive networking. A symmetric rendezvous pursuit-evasion game is developed and analyzed. Analysis and simulation results show that the modular clock algorithm performs better than random under a simple rendezvous environment model, while a modified version of the modular clock algorithm performs better than random in more difficult environment models.

Acknowledgements

I would like to dedicate this effort to my daughter, who is my daily source of inspiration and joy.

Nicholas C. Theis

Table of Contents

List of Figures

List of Tables

The Modular Clock Algorithm for Blind Rendezvous

I. Introduction

In the 21st century, wireless devices are ubiquitous and increasing in popularity. The amount of data traveling over the airways increases on a daily basis. The demands for more information to be available at higher data rates regardless of location drives up the demand for spectrum. Meanwhile, spectrum has been licensed to capacity. Spectrum has become such a precious commodity that the auction of five blocks of 700 MHz spectrum raised $20 billion dollars from big market players such as Verizon and AT&T. In addition, research has shown that certain reserved bands of spectrum are tremendously under-utilized. Radios which can leverage under-used portions of spectrum in a manner transparent to the original user would provide tremendous value. Dynamic Spectrum Access (DSA) allows secondary users to utilize unused or underutilized spectrum. In solving the spectrum shortage, advanced solutions for utilizing under-used spectrum in this manner will require advanced solutions for handling the bootstrapping of communication links.

A similar problem to spectrum scarcity is the problem of being unable to use network and communication infrastructure that has been damaged or destroyed. This is a common issue faced by first responders in crises such as Hurricane Katrina. Radios which can communicate in an ad-hoc manner and discover each other autonomously would provide the necessary adaptation required to handle these situations.

Wireless communication devices which alter their behavior based on perception and programming, known as cognitive radios, can help solve these problems. However, the lack of available network and communication infrastructure makes the initialization of cognitive radio links difficult.

As cognitive radio technology matures, the attractiveness of leveraging more flexible radios for common military problems increases. As Horine states, cognitive

radios are particularly attractive to the military sector because they require almost no support from existing infrastructure in hostile territory [9]. For these reasons and more, the paradigms of cognitive radios and DSA will become increasingly important as wireless technologies. As these radios are deployed into non-supportive or hostile environments, their ability to find each other to communicate will be tested.

Background

As the spectrum in which these radios operate becomes more dynamic, the process for the radios to find each other for communication becomes more difficult. The process to bootstrap the connection between cognitive radios is called *rendezvous*, and the means by which rendezvous is achieved may differ for every network of cognitive radios. Rendezvous in a DSA cognitive network is required when radios power up for the very first time, when the primary user arrives to evict the secondary users, or any other time the spectrum the radios were using becomes unavailable.

Much existing research has handled this problem through a variety of control structures which degraded the flexibility of the cognitive network. Control stations with reserved spectrum were required to handle the negotiation between radios operating in drastically different bands. Unfortunately, the availability of control stations and/or reserved spectrum may not exist in certain applications. In particular, many military applications of radios occur in less than hospitable environments. For this reason, we would like our devices to be able to communication equally well without pre-existing infrastructure.

Research Problem

Although the paradigm of DSA has received a considerable amount of attention, the process by which radios rendezvous and begin communicating has been largely overlooked. Most implementations simply find a method which works in a laboratory scenario. However, there are scaling concerns with many traditional infrastructure

based implementations. One of the means by which radios can overcome these concerns is by achieving rendezvous by themselves.

The problem of Blind Rendezvous (BR) exists when radios attempt to bootstrap their own connections without the use of infrastructure. How the BR problem can be efficiently solved and under what assumptions can related solutions be applied are open problems.

Scope

This thesis aims to solve the rendezvous problem without reserved infrastructure or pre-coordinated spectrum. It is assumed that communication channels are orthogonal and that the use of a channel does not impact the use of neighboring channels, as can often occur in 802.11 networks.

This thesis is not concerned with the waveforms or any other means by which the handshaking occurs once the radios arrive in common spectrum. It is also not concerned with any spectrum bargaining or allocation schemes that may occur once the radios have begun setting up their communication before data is transferred.

Approach

The nature of BR is similar to another problem in communication, which is the process of securing communications. In cognitive radio rendezvous and cryptography, the environment is unknown, hostile and there is a limited amount of information that can be leveraged. To solve these problems in cryptography, mathematical principles from number theory are used to provide a relative level of assurance. Some of the concepts used in designing cryptographic systems can be harnessed to provide assurance in the cognitive radio environment as well.

In particular, the use of modular arithmetic with prime numbers provides mathematical guarantees that can be harnessed to achieve rendezvous. These guarantees are examined against a variety of models and scenarios. Finally, the algorithms are

simulated and compared against the results from rendezvous of a purely random fashion. The objective is to demonstrate that these algorithms provide better results and guarantees than implementing a random approach.

In Chapter 2, we conduct a review of cognitive radios, DSA technology, game theory, various rendezvous problem models, and previous attempts to solve those models. In Chapter 3, we create a taxonomy for some of the more common rendezvous problem models and outline some assumptions and goals. In Chapter 4, the modular clock algorithm is defined and analyzed. A second algorithm, known as the modified modular clock, is also defined and analyzed. The design of experiments to test these algorithms and the results from the simulations are provided. Finally we conclude in Chapter 5 with the takeaways from all this analysis and simulation.

II. Background on Rendezvous and Game Theory

This chapter will describe the field of cognitive radios and intelligent network design to introduce the context in which the BR problem exists. The chapter will discuss some of the mathematical modeling techniques that have been used in other fields to introduce similar problems. Also included in this chapter are a review of other works which attempt to solve the problem and other closely related problems. Finally, since much of the theoretical literature analyzes rendezvous as a game, the chapter will examine the potential for using a game theoretic model to solve the BR and a description of different game models which can be used.

Dynamic Spectrum Access

The paradigm of DSA is to utilize available spectrum in already allocated but under-utilized frequency bands. The driving factor behind the push for DSA is the under usage of spectrum allocated for certain military, government, and public safety bands [4]. By allowing secondary users to share spectrum with primary users, we can essentially create more spectral capacity and commercial value. In order to peacefully coexist, the secondary users must be transparent to the primary users. Similarly, there is a need for secondary users to share the spectrum in a fair manner [5].

Software Defined Radios

Fette defines a Software Defined Radio (SDR) as a radio "in which the properties of carrier frequency, signal bandwidth, modulation, and network access are all defined in software [8]". He later defines a SDR as a term which abstracts underlying functionality of the applications beneath the antenna and other radio hardware. Through this methodology, the radios can be re-programmed with significantly less overhead than a purely hardware based radio.

The rapid re-programming capability of SDRs enables cognitive radios to dynamically alter and adapt their receive and transmit frequencies, signal bandwidth, and other parameters. It also allows a greater flexibility in how the signal is trans-

formed before being broadcast. This technology is a key enabler in the discussion and implementation of DSA.

Cognitive Radios

A cognitive radio is a wireless communication device which can alter its behavior based on the radio's perception of the spectrum combined with intelligence and programming. They are most frequently desired for their capabilities in spectrum management and particularly in harvesting unused portions of pre-allocated bandwidth under DSA. The term "cognitive radio" was coined by Joseph Mitola, who defined it as a smart radio which "...has ability to sense the external environment, learn from the history, and make intelligent decisions to adjust its transmission parameters according to the current state of the environment" [15]. It's important to note a bit of potentially confusing terminology present in cognitive radio literature. Silvius defines a smart radio as a type of a cognitive radio [18]. However, Mitola defined a cognitive radio as a type of smart radio. For the purposes of this thesis, the terms can be used interchangeably.

Cognitive radios leverage software defined platforms to modify their receive and transmission frequencies, change their transmission power, or classify received signals for processing. They can also leverage the existence of other cognitive radios on the network to perform joint spectrum sensing to help reduce error in the spectrum sensing problem. The collaboration capability is important yet difficult as a single node's perception of the spectrum's usage may be different than the actual usage, or the perception at any of the other nodes [9].

Cognitive Networks

A cognitive network is a collection of radios which can perform the cognitive functions of adaptation and behavior modification in response to the network operating environment. Formally, it is defined by Thomas, et al. in [16] as:

A network with a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals.

The inclusion of cognitive radios may or may not facilitate a cognitive network. Similarly, a cognitive network may exist without the inclusion of individually cognitive radios if a master controller is present. In order for communication to occur over an ad-hoc cognitive radio-based cognitive network, radios must first find each other inside of the available spectrum. If a master controller is present, the radios can communicate with the controller, which can direct cognitive radios to available spectrum. If no controller exists, the radios must find each other through the frequency rendezvous process.

Rendezvous

Rendezvous is generally defined as the process by which two players or agents meet in a common domain. The goal of a rendezvous problem is almost always to minimize the expected time required to achieve the rendezvous, called the expected Time to Rendezvous (TTR). This concept has been extended and analyzed in many different directions. The common research version of the rendezvous problem involves two agents which can move at pre-described velocities in a general planar space. Other problem variations restrict the search space to a line, a circle, or another pre-defined geometric space. Rendezvous problems and optimal solutions for minimizing TTR can differ greatly depending on the model used.

Rendezvous literature refers to the method by which agents search the available space as being symmetric or asymmetric [1]. Under symmetric rendezvous, the agents are indistinguishable and must employ the same strategy. An example of symmetric rendezvous would be writing rendezvous advice in a handbook that is provided to all hikers. Under asymmetric rendezvous, agents are allowed to use different strategies in order to minimize expected TTR. The ability to meet ahead of time to make

strategy arrangements facilitates an asymmetric approach, while the lack of ability to meet ahead of time forces a symmetric approach. One of the simpler and more well known asymmetric strategies is known as the Wait for Mommy (WFM) approach. Under WFM the "child" agent remains in their location while the "mother" agent exhaustively searches, thereby guaranteeing rendezvous. This can only be facilitated if the "mother"and the "child" are identified ahead of time.

In general, the rendezvous problem can be represented mathematically by modeling the environment as a graph $G = (V, E)$ where every vertex represents a domain for rendezvous. At each time step $t = 0, 1, 2, ...$ the agents can travel along any edge in the graph or choose to stay in place. Depending on the rendezvous model, and rendezvous is achieved either when both agents are at the same vertex and/or on the same edge at the same time.

Frequency Rendezvous

Frequency rendezvous is the process by which two cognitive radios arrive on the same frequency to begin transferring data. Silvius defines it as "the process of one smart or cognitive radio finding another in the spectrum band of interest" [19]. Some previous work on the topic refers to the process as "neighbor discovery" such as in [3], however more recent papers have taken to the term rendezvous. As DaSilva writes in [6], the rendezvous process is important in DSA in ways that go beyond the bootstrapping of communication. In DSA, the arrival of a primary user disrupts secondary communication, and the secondary users must now find each other again to resume communication.

A taxonomy for different rendezvous methods is introduced by DaSilva in [6]. As illustrated in Figure 2.1, his taxonomy consists of two branches: an aided system or an unaided system. Under an aided system, the cognitive radios contact a master radio that directs them to the same open space. This communication can either occur over a dedicated control channel or in a data channel. This system is simple to

8
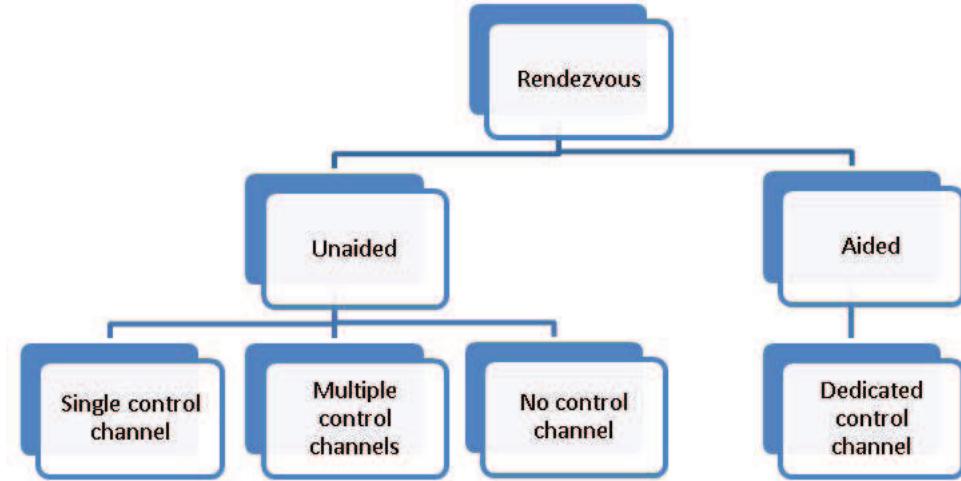
Figure 2.1: DaSilva's Taxonomy [6]

implement and provides for greater control of the spectrum, but is not very flexible or scalable. If a control channel is used, the system's bottleneck becomes closely tied to the capacity of the channel and the capacity of the controller. If data channels are used, the system is still bottlenecked by the capacity of the controller. Either way, existing infrastructure must be implemented and available in order for the cognitive radios to function.

In unaided rendezvous, radios are left to their own devices to find common spectrum. Radios may have available dedicated spectrum to use for rendezvous, facilitated by a single control channel or a series of control channels. Under a single control channel scheme, radios which seek to rendezvous contact each other over the single control channel. Under multiple control channels, radios must first find each other on one of the control channels so they can begin negotiation. If no control channel is available, the radios must figure out a way to find each other blindly, which he calls the "blind rendezvous" (BR) problem.

It could be argued that DSA schemes which require a control channel are less than dynamic. According to Sutton [21], the concept of DSA does not include a static command and control channel to direct the agents to available spectrum white space, even though contrary implementations are suggested in [4] and [5]. Control channels

can often become bottlenecks for busy networks, and make an assumption of spectrum reliability that might not always be present [7]. In order for a DSA scheme to work without a control channel, we must be able to guarantee frequency rendezvous in a reasonable amount of time.

Much of the literature on the problem assumes that radios which arrive at common spectrum will be able to find each other to begin communication. Horine's method [9] provides a means to accomplish this using slotted time windows for cognitive radios to transmit and receive in during the detection process. At the beginning of the time window, the radio senses the availability of the spectrum and emits an attention signal on the frequency of choice. If a response is received, an acknowledgement and handshake process begins. Failure to acknowledge and handshake results in starting the process over, while successful handshake concludes the rendezvous process. Although Horine suggests that the broadcast and detection process occur over many different channels, his slotting process works equally well for single channel transmission and listening. Silvius [19] describes a greater detail of beacon and handshake process, including specific message formatting. The process described would need to occur within a given time slot for successful rendezvous. If the radios are de-synchronized, this process is not likely to finish before one radio's time slot has expired.

Kowalski and Malinowski [11] define the multiple agent rendezvous problem as the gathering problem. Analysis in this paper will be done on the two agent rendezvous problem. A problem formulation with multiple agents often involves bargaining theory to reach a solution that is agreeable to all parties. For the purposes of this thesis once the agents are able to bargain, the problem is solved.

The Blind Rendezvous Problem

In order for cognitive radios to become truly autonomous, they must be able to sense and use available spectrum on their own. Furthermore, they must be able to
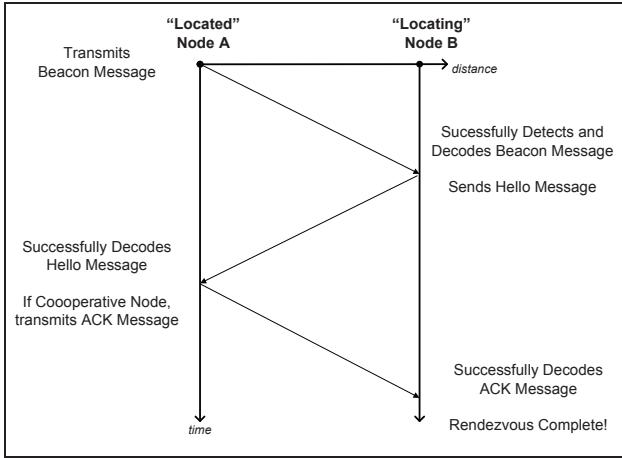
Figure 2.2: Rendezvous Handshake Process [19]

communicate with other radios without having to rely on other infrastructure or risky channel reservations, the volatility of which threatens the cognitive network. We can make this possible by solving the BR problem. The BR problem is a specific instance of the frequency rendezvous problem in which the process is unaided and no control channel or frequency allocation mechanism exists for two wireless agents to contact each other on.

Although DaSilva and Guerreiro in [7] define BR as each radio randomly visiting available spectrum, this is a specific implementation method of solving BR and not the problem itself. The BR problem is still very open for research, as most implementations for solving frequency rendezvous utilize control channels that do not scale well in large cognitive radio implementations [6]. A summary of current solutions to both the general frequency rendezvous and BR and their potential drawbacks is presented below.

Notations

The following notation is used for common variables in the rendezvous problem:

- N, the set of all cognitive radios, where $|N| = n$, and where an individual radio is given an index $i \in N$. In most formulations (and the analysis in this thesis), $n = 2$

- C_i, the set of open channels observed by the ith radio.

- $C = \bigcup C_i$ is the set of all channels being searched over by all radios

- $\bar{C} = \bigcap C_i$ is the set of open channels that radios share

- c_{ij}, the jth channel of radio i's set of open channels

- $m = |C|$, the number of available channels for all radios

- $m_i = |C_i|$, the number of open channels observed to be free by the ith cognitive radio

- $\bar{m} = |\bar{C}|$, the number of commonly observed channels

- p, a prime number

- t, a time slot variable

Existing Solutions

The following is a collection of solutions to the rendezvous problem from a wide variety of sources. Some of the sources are based in cognitive radio/cognitive network literature, some is from traditional communications literature, while others are crossovers from applied statistics and operations research.

Random Channels.   The random channel visitation is one of the more popular solutions to the BR problem. The strategy is analyzed by Balachandran in [3] and suggested for implementation by Silvius in [18]. Random channel visitation has the desirable property that rendezvous can occur with a calculable probability at any time slot t so long as $\bar{C} \neq \emptyset$.

If the available spectrum between two radios is the same ($C_i = C, \forall i \in N$, then a truly random visitation of available space will yield expected TTR in linear, or $O(m)$

time. In a more general sense, if multiple radios need to rendezvous on the same channel, then random rendezvous would have expected TTR in $O(m^{n-1})$ time. If we consider this multi-radio problem formulation as a series of pairings where two radios achieve rendezvous, with coordination they can act like a single radio then $O(log(n))$ expected rendezvous steps, and we have an expected TTR in $O(\log(n)m)$ time.

The problem with expected TTR in a random approach is that rendezvous is not guaranteed in any time. No matter what value t is chosen, there also exists a non-zero probability that rendezvous will not occur by it.

In a similar implementation, Silvius proposes the use of pseudo-random sequences in [19]. Note that the expected rendezvous time of different pseudo-random sequences would match the expected time for truly random visitation. However, there exists a probability that the same sequence is chosen by both radios with the sequence beginning at different points, resulting in a potentially orthogonal sequence. Orthogonal sequences have different values at every point of the sequence, preventing rendezvous from occurring. We would need to predefine a timeout in order to recover from these rendezvous failures under the pseudo-random sequence approach. For any value of t that could be chosen as a maximum TTR, there exists a positive probability that rendezvous will occur but has not yet by t, which makes establishing a timeout difficult.

Randomized Permutation with Wait Factor. The applied statistics world has examined the rendezvous problem at some length. Anderson and Weber's [2] symmetric discrete locations rendezvous problem formulation most closely represents the DSA rendezvous model. In this paper, agents exist within a certain discrete location (channel) at every point of time. Movement between locations is instantaneous, and occurs at the same discrete time intervals for all agents. Agents can only observe the location they are currently in, and the game ends when the two agents find each other. Furthermore, the agents are unable to see where the other player is (this is sometimes called "dark" rendezvous [1]). The agent may, however, infer that since the game

13

did not end the other agent chose a different location. In the graph formulation of Anderson's model, rendezvous must occur at a vertex and cannot occur on an edge. The graph must also be complete so that the agents can travel to any vertex at any time step.

In Anderson and Weber's paper, for a problem with m domains or vertices the algorithm/protocol by which the agents change locations consists of either following a randomized permutation of the m locations or staying put. The use of the permutation ensures that all locations are visited before a location is visited for the second time. The fact that every channel is visited in the permutation guarantees rendezvous if one agent searches while the other remains in place. The probability of choosing to remain in place for m time steps instead of following a randomized permutation for m time steps is denoted by θ. Anderson concludes that the optimal θ value, or percentage of time that a agent will stay in place for m time steps (rather than randomly permute through the channels) is 0.2475 as m gets infinitely large, and the expected TTR is approximately $0.828 \cdot m$. Note that like the random channel approach, there is no bound on the maximum TTR.

In the paper, Anderson and Weber show that the optimal solution (smallest expected TTR) for two domains ($m = 2$) is for each radio to choose between the two locations at every time step randomly. They also show that the ability to leave a message for the other agent in a visited location would result in an algorithm with a maximum TTR in $O(\sqrt{m})$ time. Unfortunately for cognitive radios, the ability to leave static messages in spectrum is not part of a standard spectrum model.

Dynamic Control Channels. In contrast to previously mentioned solutions which solve for rendezvous in an infrastructureless environment, Jeong and Yoo [10] introduce an algorithm which allows rendezvous with the base station over any available data spectrum, rather than just a specific, predefined control channel. This implementation eliminates the need for a predefined, reserved control channel between the master controller and the cognitive radios, frees up spectrum, and eliminates the

control channel spectrum bottleneck. However, the master controller (base station) itself still acts as a bottleneck, and the hardware requirements of the base towers are quite steep as it would potentially need to be listening and broadcasting over the entire dynamic spectrum.

Broadcast Rendezvous.   In [9], Horine proposes transmitting over all available frequencies sensed, and listening over the same set for responses. The cognitive radios will sense the availability of each "frequency bin" (referred to as "channel"), determine which bins are available for communication and begin broadcasting and listening. The radio collects all of the responses it receives, and then attempts to rendezvous on the "last" channel a response is received on. Transmitting and listening over multiple frequencies simultaneously (some of which may be in vastly different bands depending on the availability of channels) faces many of the same challenges in implementation as dynamic control channels such as hardware sophistication levels, antenna bandwidth, amplifier linearity, and speed.

This approach seems reasonable if only two radios exist within the spectrum, but if others are communicating in this space and the radios act without regard for the other radios, the amount of noise created by various radios broadcasting over all available spectrum could lead to a situation in which the radios begin to broadcast at higher and higher powers to ensure their signal is received over the other signals being sent. This is sometimes known as the "noisy cocktail party" effect, which can result in all radios broadcasting at their maximum power all the time. This approach can also cause interference for radios not involved in the rendezvous process.

The approach has the desirable property of being effective when the radios observe different sets of available channels as long as there exists some overlap. However, due to the number of hardware challenges and the potentially wide range of bands that would need to be broadcasted on and listened on simultaneously, it is reasonable to assume that this implementation is not currently feasible.
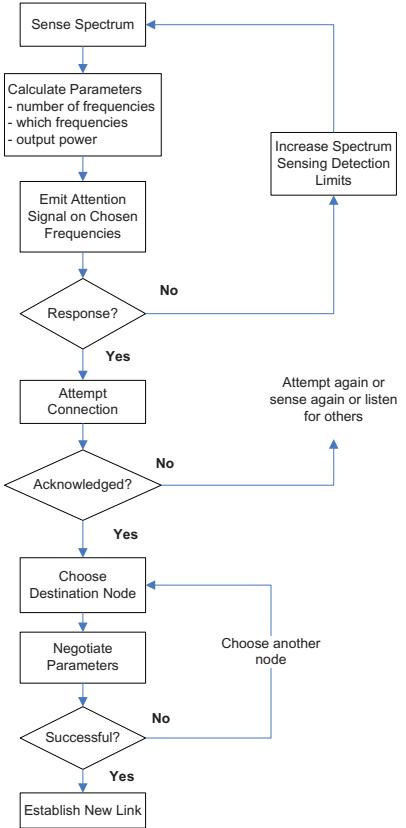
Figure 2.3: Broadcast Rendezvous Decision Tree [9]

Cyclostationary Signatures. Sutton [21] proposes the use of cyclostationary signatures to solve frequency rendezvous. A cyclostationary signature is a distinctive waveform property that is embedded in the signal over time. Their paper outlines the process to embed the signature into a transmitted signal which uniquely identifies the radio to all listeners. Radios which seek to rendezvous with the radio can then move to the indicated available space and begin communication. The strength of this approach is how it differentiates between competing secondary users in the dynamic spectrum. As dynamic spectrum networks grow in size, the need for a technique such as this to filter through the data and rendezvous noise will be necessary. However, watermarking and identifying the signal does not solve the problem of ensuring that the user of interest will ever receive the signal and actually rendezvous.

16

Generated Non-orthogonal Sequences. DaSilva and Guerreiro [7] propose the use of generated non-orthogonal channel sequences in order to achieve rendezvous. The use of non-orthogonal sequences guarantees that a rendezvous will occur regardless of the time that two radios begin searching for one another (bounding the maximum TTR). In order to create a sequence that is guaranteed to be non-orthogonal, a generalized permutation of the available channels is created and distributed amongst all radios in the network. When a radio seeks to rendezvous, it begins to execute the generated sequence. The generator provided by DaSilva works by creating a permutation of channels and then embedding this permutation within a supersequence of the permutation. Figure 2.4 illustrates this for n possible channels. For instance, assume the permutation generated and distributed to all radios is 3, 2, 5, 1, 4. The generated sequence then is 3, (3, 2, 5, 1, 4), 2, (3, 2, 5, 1, 4), 5, (3, 2, 5, 1, 4), 1, (3, 2, 5, 1, 4), 4, (3, 2, 5, 1, 4).
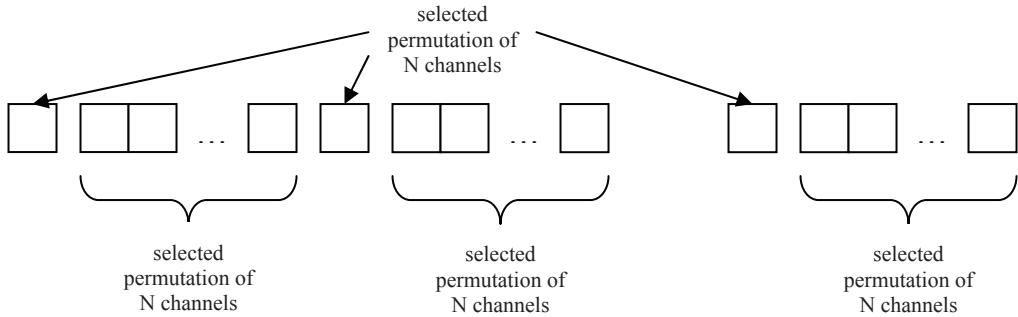
Figure 2.4: Sequence Based Rendezvous [7]

Because of the unique property of this generator, rendezvous is guaranteed to occur by the time step $t = m(m + 1)$. Asymptotically, the rendezvous process is bounded by $O(m^2)$, so as long as both radios are executing the sequence rendezvous will be achieved in squared time to the number of channels available.

This is one of the only benchmarks in open literature to date for a maximum TTR. Previous approaches only had expected TTR. This approach also has the feature that it can favor certain portions of the dynamic spectrum over others. Earlier

17

channels in the sequence are more likely to be converged on than channels later in the sequence. Unfortunately, this preference cannot be implemented by the individual radios, since both radios must use the same permutation with the generator to guarantee rendezvous.

The largest issue with the use of generated non-orthogonal sequences is that it requires pre-coordination of the generator and permutations in advance. Furthermore, both radios need to start with the same initial set of channels, i.e. $C_i = C, \forall i \in N$. As the available spectrum changes, the radios would need to receive updates, unless the sequences were implemented as a generic schema. In order to perform this pre-coordination, a control channel or out-of-band coordinator would need to be available. If the available spectrum is highly dynamic, the overhead in this communication could be quite large and possibly even larger than the overhead of directing the rendezvous in the first place. In the paper, the only non-coordinated change investigated is the dynamic removal of channels (due to presence of a primary user).

The ability of the approach to handle different sets of observed channels is analyzed further in Chapter 4.

Asymmetric Rendezvous. Silvius presents an asymmetric method similar to the WFM approach coined by Alpern in [1] for solving the frequency rendezvous problem [18]. Silvius assigns the two roles for the agents as the master radio (or "child" in WFM), which has the pre-determined responsibility of maintaining a fixed position within the dynamic spectrum, while the slave radio (or "mommy" in WFM) exhaustively searches for it.

This implementation is problematic because it requires the radios to have different roles in order to guarantee rendezvous. The situation of a master radio attempting to rendezvous with another master radio would be particularly problematic, as both radios would sit in their channel waiting for the other to find it. For very large networks and very busy nodes, a control channel on predefined spectrum would be necessary to facilitate the deconfliction of roles.

Furthermore, the solution has not been shown to be effective when the sets of observed channels differs with some overlap, (i.e. $C_i \neq C_j, \bar{C} \neq \emptyset$). In particular, we should consider the situation in which the "child" radio sits in spectrum that the "mommy" radio does not observe to be available. There exists an inherent Catch-22 for the "child" radio because if it is required to move after a timeout period, we violate our guarantee of rendezvous and our role as the waiter. However, if the "child" refuses to move from non-common spectrum, rendezvous cannot occur.

Bluetooth Rendezvous. Bluetooth communication bootstrapping uses an asymmetric approach [17] with a control channel. To initialize communication, the peripheral device initiates inquiry mode and broadcasts over the pre-defined bootstrap spectrum. Any available devices respond to the beacon messages via the control channel as well.

This is not to be confused with the fact that Bluetooth uses frequency hopping spread spectrum (FHSS) during communication to handle noise. The handling of communication once the links are created is beyond the scope of rendezvous.

Pre-defined Non-orthogonal Sequences. Some early examinations into sequences that are guaranteed to converge in less than $O(m^2)$ time and expected to converge in less than $O(m)$ time have been proposed by Martin [14]. The concept is similar in nature to the generated non-orthogonal sequences proposed by DaSilva [7], except that the exact channel sequences are calculated and given to the radios ahead of time, rather than a generator sequence. The radio observes the environment, determines the number of available channels, and begins to use the corresponding sequence. Some sample sequences are presented in Figure 2.5.

N	Sample sequence (one period)	Max TTR	E[TTR]
3	1 1 2 3 2 2 1 3 3 3 1 2	8	2.75
4	1 1 1 2 3 4 2 2 2 1 3 4 3 3 3 1 2 4 4 4 4 1 2 3	13	3.96
5	2 3 5 4 1 1 2 5 4 3 4 5 3 2 1 4 2 5 3 1 3 4 5 1 2 3 4 2 5 1	11	4.23

Figure 2.5: Pre-defined Sequence Based Rendezvous [14]

19

This approach provides greater autonomy than the previously proposed generated non-orthogonal approach, and has been calculated to have a lower expected TTR. However, there have not been any mathematical proofs to guarantee any long-term gain as the number of channels gets infinitely large. Also, the solution has not been shown to be effective when the radios observe different numbers of available channels, or if the sets of observed channels differ with some overlap. This solution also removes the ability to prefer certain channels over others.

Other Background Information

Chinese Remainder Theorem. The Chinese Remainder Theorem (CRT) states that for any set of n pairwise prime numbers $p = \{p_1, p_2, ..., p_n\}$, then for any set of integers $a_1, a_2, ..., a_n$ for which $a_i < p_i$, there exists a solution to the set of equations

$$x \equiv a_1 \mod (p_1)$$
$$x \equiv a_2 \mod (p_2)$$
$$...$$
$$x \equiv a_n \mod (p_n)$$

The CRT is important to this research because it proves that there exists a solution to the set of equations, and regardless of which values of a_i are desired. Our main concern in applying the CRT is that the moduli are pairwise prime, so that regardless of the starting integers given we will have a solution. The use of absolute primes guarantees that the set P will also be pairwise co-prime. If the numbers chosen were not absolute primes, coordination would need to occur to ensure that no common factors exist between chosen p values.

Fundamentals of Game Theory. According to Osborne and Rubenstein [13], game theory is "a bag of analytical tools designed to help us understand the phenomena that we observe when decision-makers interact." As we seek to increase the automation and intelligence of our network service providing agents, the potential for a conflict between the goals of the network and the goals of the individual agent

increase. Game theory can be used to help us understand how those interactions play out over time.

In order to model a scenario as a normal form game G, where $G = \langle N, A, U \rangle$ the following components must be represented:

1. A set of 2 or more players (or radios) $N, |N| = n$

2. A set of actions for player i, A_i. The set of all actions possible is referred to as the action space A, where $A = A_1 \times A_2 \times \ldots \times A_n$.

3. A set of utility or objection functions for each player u_i, where $u_i : A \to \mathbf{R}$.

It's important to note a particular game theory notation where i refers to any particular player, and $-i$ refers to every other player except player i. Formally, we represent this as $i \in N, -i = \{N - \{i\}\}$.

As an example, consider the following simple 2 person game, represented in normal form. The players are at a picnic with a limited amount of food and must choose between chicken or beef to eat. If both players choose the same meat, they wind up not getting enough to eat and are dissatisfied. Player 1's action choices are given by {Chicken, Beef} and the decisions is represented by the row in Table 1.1. Player 2's decision is represented by the column. In each of the spaces in the table is the utility score for each player in the form (P1, P2). For the rows and columns where the decisions match each other, the players have chosen the same meat and would therefore be dissatisfied, reflected by their utility of 0. If the players choose different meats, they receive a much better utility. Notice that the player that eats chicken alone has a higher utility function than the player that eats beef alone, but that both players would prefer eating beef alone than sharing meat.

Table 2.1: A very simple game

		P2	
		Chicken	Beef
P1	Chicken	(0, 0)	(5, 2)
	Beef	(3, 6)	(0, 0)

Strategies. At a high level, a strategy is a player's plan to guide which actions they will choose based on the scenarios presented to them throughout the various iterations of a game. Strategies are considered finite if the actions spaces are also finite and the game is not repeated or of infinite stages, i.e. *finite game*, as is the case in the game presented in Table 1.1. Under a pure strategy, the player pre-defines which moves he will make ahead of time based on the situation they are currently in. Alternatively under a mixed strategy, the player will choose their strategy based on some probability function.

Often times, strategies for simple games can be compared to each other by considering their dominance. It is said that a strategy A strictly dominates strategy B if choosing A always presents the player with a superior outcome no matter what the other players do [13]. Similarly, a strategy A weakly dominates strategy B if at least one outcome of A is better, and the rest are no worse. For strict dominance, formally:

$$\forall s_{-i} \in S_{-i}, u_i(A, s_{-i}) > u_i(B, s_{-i}) \tag{2.1}$$

Alternatively, strategies are intransitive and their success depends on the strategies of the other players. Some commonly represented multi-stage or repeated game strategies are:

- Grim Trigger - Players will choose mutually beneficial actions until another player deviates. Once someone else deviates, the player will choose greedily.

- Tit for Tat - Players will reciprocate their opponent's action.

- Collusion - A subset of the players agree to choose actions that will benefit themselves, potentially at the cost of other players. Collusion is typically facilitated through external communication.

Nash Equilibrium. At the points where each player is eating a different meat, neither player can improve their utility function by individually altering their decision. This is known as a Nash Equilibrium (NE). Every finite game in normal

form has at least one NE under mixed strategies [13]. The game above has two separate NE– (Chicken, Beef) and (Beef, Chicken). At either equilibrium, if either player unilaterally switches their choice of meat they will arrive at (Chicken, Chicken) or (Beef, Beef) which results in a utility of 0, which means they cannot unilaterally improve their utility. These equilibriums are used to predict the results of the game as the greedy behavior of the agents will lead them to choose better solutions for themselves until they are unable to do so. A NE for a sequential or one shot game is formally defined as an action vector $a \in A$ where for any player i and any alternate strategy a_i':

$$u_i(a) \geq u_i(a_i', a_{-i}) \tag{2.2}$$

In another sense, if we consider the game to be a function with the players' decisions as inputs, the NE are local maxima of the function. NE do not answer the question of how the game arrived at one particular NE over another, nor does predict if a NE exists, if it is unique, and in the case which it's not, which one will be arrived at. This can present system designers with challenges as they attempt to create efficient systems from autonomous agents.

Pareto Optimality. Ideally, we would like our equilibria to be Pareto Optimal (PO), which is a solution where there is no other outcome in A where one player can improve their utility while the other players do no worse [20]. In other words in a PO solution nobody can improve without making the other players suffer. Formally, we define an action vector $a \in A$ as PO if:

$$\nexists b \in A, u_i(b) \geq u_i(a) \forall i \land \exists j, u_j(b) > u_j(a) \tag{2.3}$$

Unfortunately, we cannot guarantee that a NE solution is PO, nor can we guarantee that a PO solution is a NE. Pareto optimality is often used as a measurement for the quality or efficiency of an outcome, and particularly a NE. However, just because a solution is PO does not guarantee that it is fair. Although the players do their best to

individually improve their utility, there might be game or system-wide goals of Pareto optimality that can require coordination to be achieved.

Coordination Game. In a coordination game, the players must work together to achieve their maximum utility. These games are some of the more popular games for evaluation as they often used for studying economics. An classic example of a coordination game is known as the Stag Hunt. In this game, the players can either choose to hunt for a stag or a rabbit. Stags make a much more satisfying meal than a rabbit does. However, stags are very difficult to hunt and require two people to hunt successfully.

Table 2.2: Stag Hunt

| | | P2 | |
		Stag	Rabbit
P1	Stag	(10, 10)	(0, 3)
	Rabbit	(3, 0)	(3, 3)

Zero-Sum Game. A zero-sum game is a class of non-cooperative games where the total utility for the system remains at zero regardless of the actions chosen by the players. In these games, players can only increase their utility at the expense of the other players. By the zero-sum property, all solutions to a zero-sum game are PO.

Pursuit and Evasion Game. A pursuit-evasion game is a specific type of zero-sum game with a variable termination time. Depending on how the game is set up, the pursuer gains utility as its strategy gets closer in the action space to the evaders. Similarly, the evader gains utility by being sufficiently far away from the pursuer.

In pursuit-evasion games, it's important to define whether the evader can be "captured" or not. For the purpose of considering rendezvous with cognitive radios, we think of the pursuer as a jammer rather than a captor. Therefore, radios cannot be captured and are free to change their frequencies to evade jamming, however they

suffer a heavy penalty for having been jammed through lost time during the jamming along with lost time finding another channel to communicate on with the distant end.

Rendezvous Pursuit-Evasion Game

W. Lim's paper [12] models the rendezvous problem as a multi-stage pursuit-evasion game with three players. R is a set of two rendezvousing agents $\{R_1, R_2\}$, and S is the capturer. These three players can exist in any one of m locations. The payoff of the game for S is 1 and 0 for $\{R_1, R_2\}$ if S and either R_1 or R_2 occupy the same playing space at the same time. In other words, the evading agents can be "captured" in this game. If R_1 and R_2 occupy the same space first, the payoff for S is 0 and the payoff for $\{R_1, R_2\}$ is 1. If neither occupy the same space, the game is a draw and the utility for S, R_1, R_2 is 0.

Table 2.3: Partial Three Player Pursuit-Evasion Game Utility Table for $m = 3$

R_2 chooses location 2		S		
		Location 1	Location 2	Location 3
	Location 1	(0, 0, 1)	(0, 0, 1)	(0, 0, 0)
R_1	Location 2	(1, 1, 0)	(0, 0, 1)	(1, 1, 0)
	Location 3	(0, 0, 0)	(0, 0, 1)	(0, 0, 1)

Table 2.3 presents a normal form representation of Lim's game. Note that the respective payoffs tuplets represent the payoffs for (R_1, R_2, S) respectively. By giving draws a utility of .5, we preserve the agents' and capturer's motivations to find more optimal solutions.

Therefore, we consider this game as non-cooperative between the agents R and the capturer S. Lim notes that if a random strategy were used by all players, all meetings are equally likely and the expected utility for S is $\frac{2}{3}$ $\forall m$.

Lim begins the game by starting R_1, R_2, and S in random but separate locations, which they locally designate as $R_1(1)$, $R_2(1)$, $S(1)$. Since no two players occupy the same space, $u = (0, 0, 0)$. For a simple game where $n = 3$, knowing that they begin

25

in separate locations is important because every room is occupied at the start. The agents (R_1, R_2) use the following asymmetric multi-stage mixed strategy:

- Stage 0. R_1 is in location $R_1(1)$ and R_2 is in location $R_2(1)$. $R_1(1) \neq R_2(1)$.

- Stage 1. R_1 moves to $R_1(2)$, randomly choosing between the two locations it hasn't visited, while R_2 stays in its starting room $R_2(1)$.

- Stage 2.

    Type 1: With probability $\frac{7}{19}$, R_1 goes to the only location it hasn't visited while R_2 stays put.

    Type 2: With probability $\frac{6}{19}$, R_1 returns to its starting location $R_1(1)$ while R_2 moves to one of the two locations it hasn't visited $R_2(2)$.

    Type 3: With probability $\frac{6}{19}$, R_1 stays at its current location, $R_1(2)$, while R_2 moves to one of the two locations it hasn't visited, designated $R_2(2)$.

- Stage 3+. Randomly choose from the three locations with probability $\frac{1}{3}$.

Furthermore, the capturer S chooses the following multi-stage mixed strategy:

- Stage 0. S is in location $S(1)$. $S(1) \neq R_1(1)$, $S(1) \neq R_2(1)$.

- Stage 1. Move to a new location $S(2)$.

- Stage 2.

    Type 1: With probability $\frac{5}{19}$, return to location $S(1)$.

    Type 2: With probability $\frac{5}{19}$, stay at location $S(2)$.

    Type 3: With probability $\frac{9}{19}$, move to the last location $S(3)$.

- Stage 3+. Visit each location with probability $\frac{1}{3}$.

Note that the only way the initial game is not over after Stage 1 is the case in which R_1 and S trade places, which will occur $\frac{1}{4}$ of the time. Utilizing the mixed strategies previously outlined, Lim proves that the expected utility for S has an upper

bound of $\frac{47}{76}$, which is approximately 93% of $\frac{2}{3}$, the expected utility for S if $\{R_1, R_2\}$ employ a random strategy.

For $n > 3$, Lim assumes that the asymmetric WFM approach from Anderson and Weber is followed by $\{R_1, R_2\}$. Under the WFM approach, one agent will remain in place (R_1) while the other agent (R_2) follows a randomized permutation of all n locations. Lim states that S cannot do better than to use the same strategy as R_2. He then proceeds to demonstrate that $1 - e^{-1}$ is the upper bound for S's expected utility with a large n. He demonstrates this by examining the randomized permutations of R_2 and S and calculating the probability that R_1's location is earlier in S's permutation than in R_2's permutation. The probability of S finding R_1 first combined with the probability of S finding R_2 while searching gives us the expected utility for S. Lim also states that the WFM strategy is not optimal for $\{R_1, R_2\}$, but offers only a very minor revision.

Lim's paper is useful because it defines a game which can be used to analyze the rendezvous process when agents can be captured. The limitation of the approach is that it relies on information from the number of channels being small. It also relies on an asymmetric approach to solve the problem, which requires coordination from the agents ahead of time. The paper also does not give the reader any insight as to how the mixed strategy for $m = 3$ was derived, or any analysis into whether it might be optimal.

III. Modeling the Blind Rendezvous

In considering how to solve the Blind Rendezvous problem, it's important to define the environment that the solution will operate in. This chapter enumerates many common factors in DSA cognitive networks which can affect the process of rendezvous. It also lists the assumptions made by this thesis to analyze algorithm performance, and the goals of the endeavor. Furthermore, the chapter will outline and describe the model used to create the modular clock algorithm.

Problem Variables

Along with the factors which define the rendezvous methods introduced in Figure 2.1, this thesis introduces a taxonomy for cognitive network models and how they relate to the rendezvous problem. The following is a non-exhaustive list of variables and system requirements which can be used to define the rendezvous model:

1. Timing between radios (i.e. slotting)

   Radio synchronization affects the ability to initialize communication. Certain models require in-depth strategies to deal with any level of synchronization.

2. Symmetric or asymmetric (role based) strategies

   As mentioned in Chapter 2, the use of roles in a model can reduce expected TTR. However, the assignment of roles can be difficult.

3. Number of radios

   The number of radios can affect whether a rendezvous solution is complete or incomplete after initial rendezvous. In the case of only two radios, any rendezvous solution is complete.

4. Presence of common spectrum naming

   For any unique physical frequency channel, the identification of the channel for each radio may be dissimilar. This is problematic when considering order based rendezvous schemes as the naming affects the mathematical model used.

5. Ability to perform wide-band sensing

   The ability for a single radio to sense and/or transmit in multiple frequencies. Being able to do so simplifies rendezvous, but there are complications with hardware costs and noise.

6. Policy of frequency fairness

   Fairness is to consider every available channel with some probability for rendezvous.

7. Existence of one or more control channels

   Control channels are reserved channels that are only used to facilitate rendezvous.

8. Presence of a master controller

   Master controllers are central agents which can control spectrum allocation for the radios in a cognitive network.

9. Amount of common spectrum

   Increasing the amount of common spectrum improves the radios' ability to find common spectrum and therefore rendezvous.

10. False Detection of Primary User

    In DSA networks, the arrival of a primary user causes the secondary users to vacate the channel. If radios falsely detect a primary user, it can affect their ability to rendezvous and decrease the amount of common spectrum.

11. Jamming/Malicious Activity

    Malicious activity can resemble a primary user or otherwise make less spectrum available to the radios for rendezvous.

Considering these factors, we categorize models as used in rendezvous literature. We use the names of the authors who described the model as nomenclature for the model.

In Table 3.1, we introduce the common model. This model represents one of the more traditional methods of implementing a DSA cognitive network. It combines the use of a control channel with a master controller to handle all connection rendezvous. The following model does not represent a situation where a blind rendezvous algorithm is necessary.

Table 3.1: Common Model Assumptions

Variable	Model Assumption
Radio Timing	No slotting
Roles	Asymmetrical
Number of Radios	Any number
Common Spectrum Naming	Channels are commonly named
Wide-band	No wide-band sensing or broadcasting available
Fairness	Available spectrum analyzed and assigned by master controller rather than radios
Control Channels	Control Channel
Master Controller	Master controller
Common Spectrum	All radios sense the same available spectrum as the master controller or some subset
False Detection	No false detection
Malicious Activity	No malicious activity

Table 3.2 describes Jeong and Yoo's model [10], which is very similar to the common model except the set of pre-defined control channels is removed.

The Silvius model in Table 3.3 was designed to be more agile and robust for first responders at an emergency scene. As such, the model is one of the first to formally abandon the approach of using any reserved infrastructure. In comparison to the Jeong model of Table 3.2, the Silvius model abandons the master controller altogether. Instead, rendezvous is facilitated through asymmetric means.

In his work in [6] and [7], DaSilva makes a strong case for moving away from the common model towards a more dynamic model. Noting the lack of scalability in the Common model and the Silvius model, DaSilva proposes one of the first models which uses a symmetric rendezvous approach with no infrastructure to achieve rendezvous.

Table 3.2: Jeong Model Assumptions

Variable	Model Assumption
Radio Timing	No slotting
Roles	Symmetrical
Number of Radios	Any
Common Spectrum Naming	Channels are commonly named
Wide-band	Wide-band sensing necessary for the master controller
Fairness	Fairness determined by master controller
Control Channels	No control channels
Master Controller	Master controller directs all rendezvous
Common Spectrum	All radios sense the same available spectrum as the master controller
False Detection	No false detection
Malicious Activity	No malicious activity

Table 3.3: Silvius Model Assumptions

Variable	Model Assumption
Radio Timing	Radios are loosely slotted
Roles	Asymmetrical
Number of Radios	Two
Common Spectrum Naming	No common naming
Wide-band	No wide-band sensing or broadcasting available
Fairness	All channels must be considered equally for rendezvous
Control Channels	No control channels
Master Controller	No master controller
Common Spectrum	All radios sense the same available spectrum
False Detection	No false detection
Malicious Activity	No malicious activity

The symmetric approach in table 3.4 removes the need to facilitate the roles from the Silvius model.

The model proposed by DaSilva in [7] very closely resembles the model used by Anderson and Weber in [2], shown in Table 3.5. The major distinctions between the two are their assumptions of synchronization and common channel names. Although they come from the field of Applied Statistics, Anderson and Weber wrote one of the first papers on rendezvous in discrete locations. In the paper, the problem is phrased

Table 3.4: DaSilva Model Assumptions

Variable	Model Assumption
Radio Timing	Radios are loosely slotted
Roles	Symmetrical
Number of Radios	Two
Common Spectrum Naming	Channels are commonly named
Wide-band	No wide-band sensing or broadcasting available
Fairness	All observed channels must be considered for rendezvous
Control Channels	No control channels
Master Controller	No master controller
Common Spectrum	All radios sense the same available spectrum
False Detection	No false detection
Malicious Activity	No malicious activity

as two players seeking to meet in m unique discrete locations which can be traveled between instantaneously. Alpern [1] modified the problem description without changing the model by calling it the "telephone game." Under Alpern's problem description, two players in two separate locations have m telephones directly connected to each other. At specific time intervals, both players choose a telephone, pick it up, and say "hello." The game ends when the players choose the same phone in the same time slot and can have a conversation. Although the papers were phrased in more general game theoretic terms, their players have been renamed radios for the purpose of the model. Many of the assumptions made were for the simplification of mathematical analysis.

Finally, Table 3.6 introduces a model that makes the fewest assumptions about the environment, known as the free-for-all model (FFA). Under this model, there exists no infrastructure to assist the radio in finding other radios, nor do we know anything about the availability of spectrum or the hostility of the environment. Rendezvous in this environment can be very difficult.

Table 3.5: Anderson and Weber Model Assumptions

Variable	Model Assumption
Radio Timing	Radios strictly slotted
Roles	No assumption of roles, but different solutions for symmetric and asymmetric
Number of Radios	Two
Common Spectrum Naming	Channels are commonly named
Wide-band	No wide-band sensing or broadcasting available
Fairness	All channels considered with equal probability
Control Channels	No control channels
Master Controller	No master controller
Common Spectrum	All radios sense the same available spectrum
False Detection	No false detection
Malicious Activity	No malicious activity

Table 3.6: Free-for-all Model Assumptions

Variable	Model Assumption
Radio Timing	No slotting
Roles	Symmetrical
Number of Radios	Any number
Common Spectrum Naming	No common naming
Wide-band	No wide-band sensing or broadcasting available
Fairness	No rule for fairness
Control Channels	No control channels
Master Controller	No master controller
Common Spectrum	No amount of common spectrum assumed except that some exists
False Detection	False detection of primary user possible and likely
Malicious Activity	Malicious nodes interfere with radios, spoofing primary users

Assumptions

Table 3.7 describes the assumptions made in this thesis in researching and developing a process for BR in cognitive networks. The model assumed by Anderson and Weber most closely represents the model used, except that the modular clock model makes only the assumption that there exists some common spectrum.

Table 3.7: Modular Clock Model Assumptions

Variable	Model Assumption
Radio Timing	Loosely slotted so that two radios which choose the same spectrum at the same time step are guaranteed to rendezvous
Roles	Symmetrical, as the use of asymmetric rendezvous could create radio pairings which are unable to rendezvous
Number of Radios	Two
Common Spectrum Naming	No common naming
Wide-band	No wide-band sensing or broadcasting available
Fairness	All channels must be considered equally for rendezvous
Control Channels	No control channels available, as control channels serve as a bottleneck for the cognitive network
Master Controller	No master controller available. Similar to control channels, master controllers serve as a bottleneck for the network both in bandwidth, size, and distance
Common Spectrum	The amount or percentage of common spectrum is not assumed except that some common spectrum between the two radios must exist
False Detection	No assumption made. If communication between two radios is interrupted for any reason, the rendezvous process must begin
Malicious Activity	No malicious activity considered for algorithm development and analysis

Goals. The following are the goals outlined in the creation and analysis of a Blind Rendezvous algorithm.

1. If all spectrum is common, achieve rendezvous in guaranteed fewer than m^2 iterations. The metric of m^2 was chosen due the upper bound of $O(m^2)$ convergence time by DaSilva [7].

2. Perform rendezvous without any prior communication or central architecture. Only the algorithm or process need be known by each radio.

3. Achieve rendezvous in finite time with non-zero probability so long as $\bar{C} \neq \emptyset$.

34

Model Difficulty

Discussing difficulties of models can be very subjective. The more *a priori* knowledge we have about the environment, the easier the problem becomes. Lack of knowledge or assumptions can make the problem very difficult. The amount of simplifying assumptions made reduces the amount of variation in the rendezvous problem. The following figure is a qualitative estimation of the relative difficulty of the models. Models are labeled as easier or more difficult depending on the amount of *a priori* knowledge and simplifying assumptions made regarding the environment.
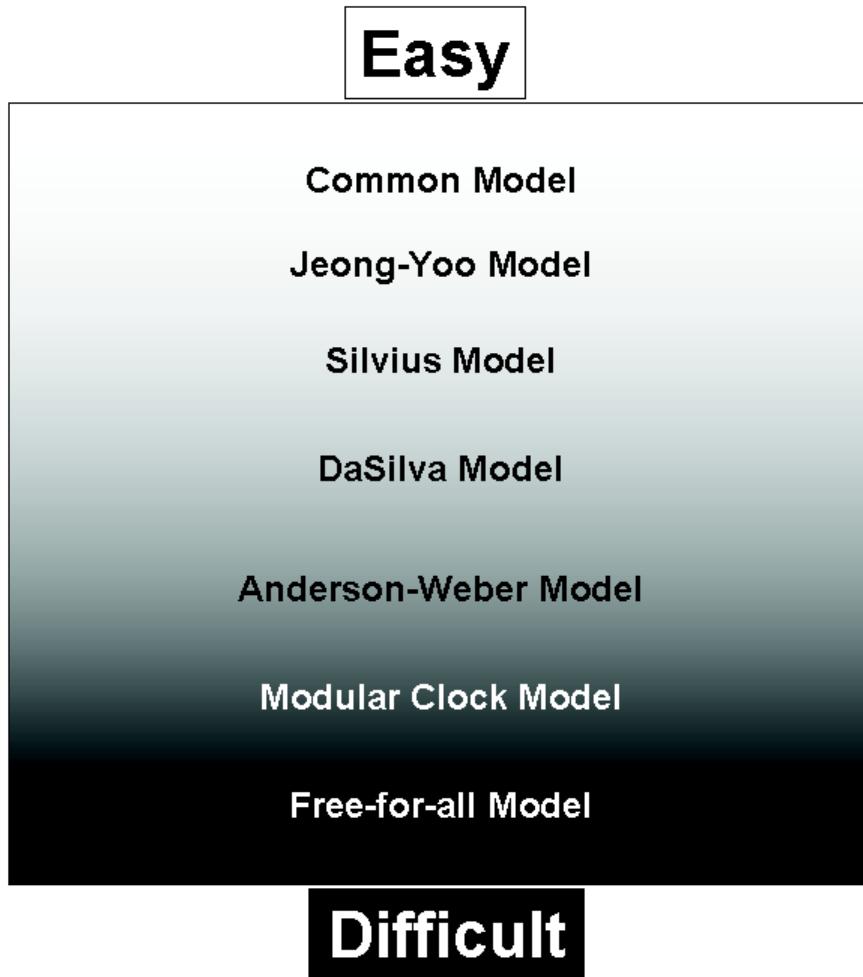
Figure 3.1: Synopsis of model difficulties

IV. The Modular Clock Algorithm and Other Analysis

This chapter will take a deeper look at the flexibility of the rendezvous process described by DaSilva in [7]. It will take a cursory look at some simple Blind Rendezvous games and compare them to the game proposed by Lim in [12]. It introduces and analyzes the modular clock algorithm to solve the Blind Rendezvous problem under the DaSilva model, proposes an alternate implementation of the algorithm under the modular clock model, and then provides statistical analysis of both algorithms against random rendezvous.

Analysis of Generated Non-orthogonal Sequences

In considering the best option for choosing methods for frequency rendezvous, the generated non-orthogonal sequences proposed by DaSilva in [7] currently provide the best known absolute guarantee of maximum rendezvous time. However, a number of assumptions presented must be evaluated further to truly understand whether the algorithm is robust enough for implementation.

Rendezvous under different labeling. As a first attempt to break the generated non-orthogonal sequences, consider running them under the modular clock model, in which two radios share a common space for the channels, but do not refer to the same channels by the same name. In this example, consider a set of 5 channels with universal labels A through E. For first radio, it will locally label A as 1, B as 2, C as 3, D as 4, and E as 5. For the second radio, it will locally label A as 5, B as 1, C as 2, D as 3, and E as 4. The shared permutation provided is $3, 2, 1, 4, 5$. Therefore, the first radio's permutation under the universal labels is C, B, A, D, E, while the second radio's permutation is D, C, B, E, A.

It is easy to see the sequences can become orthogonal. Take the case in which the sequences start at the same time: there is no point at which both radios are on the same channel in the same slot. At every point, both radios will choose the same local label, which map to different physical channels.

Master Permutation	3	2	1	4	5
Radio 1	C	B	A	D	E
Radio 2	D	C	B	E	A

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Radio 1	**C**	C	B	A	D	E	**B**	C	B	A	D	E	**A**	C	B	A	D	E	**D**	C
Radio 2													**D**	D	C	B	E	A	**C**	D

t	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Radio 1	B	A	D	E	**E**	C	B	A	D	E	**C**	C	B	A	D	E	**B**	C
Radio 2	C	B	E	A	**B**	D	C	B	E	A	**E**	D	C	B	E	A	**A**	D

t	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
Radio 1	B	A	D	E	**A**	C	B	A	D	E	**D**	C	B	A	D	E	**E**	C
Radio 2	C	B	E	A	**D**													

Figure 4.1: Rendezvous failure under different channel names

More generally, this problem arises when both sequences are on their super-sequence at the same time, as shown in Figure 4.1. Figure 4.1 demonstrates the scenario in which we begin the second radio's sequence 12 time steps after the first radio begins. Note that at time 42, we've gone $m(m+1) = 30$ time steps without rendezvous.

However, if we consider the situation in which the second sequence begins in the middle of the subsequence of the first radio, we find that rendezvous will still occur by $m(m+1)$ time steps. If the supersequence of Radio 2 begins during the subsequence of Radio 1, there exists an crossover point in which the channel for Radio 1 remains the same while the channel for Radio 2 rotates, as shown by the highlighted areas in Figure 4.2.

The result of this analysis is that while the sequence is still likely to converge, it is not guaranteed to. If a generated non-orthogonal sequence is implemented and channel names are non-global, there would need to be some sort of timeout mechanism or "hiccup" factor introduced to jar orthogonal sequences out of alignment. This may be handled by introducing a randomized wait factor after each sequence.

37

Master Permutation	3	2	1	4	5
Radio 1	C	B	A	D	E
Radio 2	D	C	B	E	A

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Radio 1	C	C	**B**	A	D	E	B	C	**B**	A	D	E	A	C	**B**	A	D	E	D	C
Radio 2			**D**	D	C	B	E	A	**C**	D	C	B	E	A	**B**	D	C	B	E	A

t	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Radio 1	**B**	A	D	E	E	C	**B**	A	D	E	C	C	**B**	A	D	E	B	C
Radio 2	**E**	D	C	B	E	A	**A**	D	C	B	E	A	**D**					

Figure 4.2: Rendezvous still guaranteed sometimes under different channel names

Analysis under channel addition. In the original paper, DaSilva proves that the algorithm can handle the dynamic arrival of primary users. In a DSA scheme, this is very important for our cognitive radios. However we must consider whether the algorithm can handle the dynamic departure of primary users which adds available channels. Unfortunately the paper does not indicate how the departure of a primary user would be handled, so here some methods are suggested.

First we consider this under the DaSilva model, in which the channel sets observed by the radios are the same. As channels become available, we begin the generated sequence based on the new set of available channels. Since we have no controller to direct us to a new permutation as the available channels change, we will need to know the permutation for every number of channels we could observe. This implementation could have a very large overhead, require a large lookup table of sequences and could hinder rendezvous if arrival and departure occur frequently. Alternatively, a general ordered sequence could be used, or the sequence could contain all possible channels in a master permutation.

Consider the implementation of the latter option – a master permutation which contains the exhaustive list of all possible channels, and the radios remove the channels they do not see available. The master permutation would have length m, where

$m = max(|C_i|)$. Consider when $m = 10$ and the channels are given labels A through J, and the master permutation E, J, I, C, G, A, D, B, H, F. Under the DaSilva model, if both radios sense the availability of channels B through G and J, they would create the sequence E, J, C, G, D, B, F. If the channel A suddenly became available, both radios would know to add A between G and D. Since they would still be using the same sequence, the guaranteed properties of rendezvous will still hold.

However, consider the situation in which a new channel would need to be added to the master permutation. Without any outside knowledge, the radios would have a $\frac{1}{m+1}$ chance of adding the $(m + 1)$th channel in the same spot as the other radio. In order to guarantee the channel lists remain the same, coordination between all radios would be necessary. This would be very difficult to implement as the addition of a channel to the exhaustive list would require reconfiguring all of the radios in the network.

If the departure of a primary user is handled in a mutually agreed upon means, the radios will wind up using the same permutation. As long as the radios are using the same permutation on the same channel set, they are guaranteed rendezvous.

Analysis under modular clock model. Although the original DaSilva paper makes no claim of being able to handle rendezvous when the two radios observe different channel sets, its ability to do so must be evaluated when being considered for implementation.

Using the modular clock model, guaranteed rendezvous by $O(m^2)$ time steps can be shown to fail by example. In the case where both radios observe a different set of channels, but the same number of channels ($m_i = m_j, C_i \neq C_j$), we have an analogous scenario as rendezvous under different labels, previously discussed.

In the case where both radios observe a different number of channels ($m_i \neq m_j$), we can demonstrate there will be no rendezvous in $m(m + 1)$ time (even where $m = max\{m_i, m_j\}$). Consider the case in which $C_1 = \{1, 2, 3\}$ and $C_2 = \{3, 4, 5, 6, 7, 8, 9\}$ as represented in Figure 4.3. Assume a common labeling for channels and the per-

mutations are in numerical order, $m = max\{3, 7\} = 7$. Beginning the sequences at the same time, there is still no rendezvous after $7 \cdot (7 + 1)$ or 56 time steps. This demonstrates the loss of guaranteed rendezvous in $m(m + 1)$ time steps.

Master Permutation	1		2		3		4		5		6		7	
Radio 1	A		B		C									
Radio 2	C		D		E		F		G		H		I	

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Radio 1	A	A	B	C	B	A	B	C	C	A	B	C	A	A	B	C	B	A	B	C
Radio 2	C	C	D	E	F	G	H	I	D	C	D	E	F	G	H	I	E	C	D	E

t	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Radio 1	C	A	B	C	A	A	B	C	B	A	B	C	C	A	B	C	A	A
Radio 2	F	G	H	I	F	C	D	E	F	G	H	I	G	C	D	E	F	G

t	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
Radio 1	B	C	B	A	B	C	C	A	B	C	A	A	B	C	B	A	B	C
Radio 2	H	I	H	C	D	E	F	G	H	I	I	C	D	E	F	G	H	I

Figure 4.3: Rendezvous failure with different channel sets

Generated non-orthogonal sequences provide the best guaranteed time of any approach studied in this thesis and are guaranteed to work under the DaSilva model. However, the properties of guaranteed rendezvous can fail under modular clock model conditions, which can occur in cognitive networks.

Game Theory for Blind Rendezvous

As mentioned in chapter 2, game theory is a good mathematical basis for understanding the decision process that cognitive radios will go through. With that in mind, we examine some game theoretic models for BR under models that do not use infrastructure, such as the DaSilva model, the Anderson and Weber model, and the Free-for-all model.

Simple Blind Rendezvous Game. The simplest representation of game theory as a game is to break up the available spectrum into m numerically sequenced chan-

nels, $C_i = (c_{i1}, c_{i2}, ...c_{im})$ under the DaSilva model. If we consider the channels as the action space for the radios, we have a coordination game in which each radio receives utility if their action matches the other radio's action, and none otherwise. For the purposes of simplification, the model only considered two radios.

Table 4.1: Simple Blind Rendezvous Coordination Game

		P2				
		Ch 1	Ch 2	Ch 3	...	Ch m
	Ch 1	(1, 1)	(0, 0)	(0, 0)	...	(0, 0)
	Ch 2	(0, 0)	(1, 1)	(0, 0)	...	(0, 0)
P1	Ch 3	(0, 0)	(0, 0)	(1, 1)	...	(0, 0)
	...	...	...	...	...	...
	Ch m	(0, 0)	(0, 0)	(0, 0)	...	(1, 1)

As would be expected, the NE of the system are the points in which the radios have rendezvoused, and all rendezvous solutions are PO. Given an arbitrary starting channel, there's no rendezvous which would be arrived at more often than others. If the two players choose the same channel, their utility will be observed by initiating the communication. If the two players choose different channels, they receive no utility as they are unable to communicate. Furthermore, the radios cannot observe the action (channel selected) of the other radio and utility gives no insight into the state of the game. Therefore, players cannot gain knowledge in a repeated game by which to improve their expected utility, making strategies that guide future actions is difficult. For this formulation, a mixed strategy of selecting any channel with $\frac{1}{m}$ probability is a good strategy.

Proximity Blind Rendezvous Game. Next, the DaSilva model was slightly tweaked, allowing radios to sense adjoining channels with some utility as an inverse square to the distance from the other radios. The justification for this utility function was to attempt to model inter-channel interference that might be detectable if radios broadcast in neighboring channels to each other and filter roll-offs are not adequate.

The utility function u_i for this game was:

$$u_i = \frac{1}{1 + (a_i - a_{-i})^2} \qquad (4.1)$$

This utility improves as the radio moves closer to other radios. Given a utility, a radio can tell how close it is to its nearest neighbor. However, it does not know which neighbor it is close to, or what spectrum the radio is in. Since there were two radios, each will receive the same utility. As such, only one utility value is written in each of the spaces. The utility table for $m = 5$ is shown in Table 4.2.

Table 4.2: Blind Rendezvous Proximity Coordination Game

		P2				
		Ch 1	Ch 2	Ch 3	Ch 4	Ch 5
	Ch 1	1	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{17}$
	Ch 2	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{10}$
P1	Ch 3	$\frac{1}{5}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{5}$
	Ch 4	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{1}{2}$	1	$\frac{1}{2}$
	Ch 5	$\frac{1}{17}$	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{1}{2}$	1

An interesting facet of this game is that the average utility in the middle rows and columns is greater than that of the outer rows and columns. This makes selecting channels in the middle of the available spectrum more attractive as a strategy, because the player can minimize the maximum distance to the opposite radios.

This game is not without its limitations. The precise allocation of channels in spectrum prevents much physical bleedover into other channels. While it may be physically possible to detect activity in a neighboring channel, any detection beyond neighboring channels would be quite difficult. It's possible that under the free-for-all model, this approach can be more useful because the channels aren't nearly as well defined and monitoring weaker signals can be useful.

Proximity and Preference Blind Rendezvous Game. The previous game was improved by considering a preference relationship between the radio and a particular channel. Keeping in theme with the previous experiment, the model was modified by providing two factors of utility: One for the distance from the other radio and one for the distance from a channel of preference, c_{i*}. This preference might exist because of a good signal-to-noise ratio, or because the radio perceives infrequent use by a primary user. For the purpose of this game, the reasons behind the preference are not considered. Again, only two radios were considered, the utility function is:

$$u_i = \frac{1}{1 + (a_i - a_{-i})^2} + \frac{1}{1 - (a_i - c_{i*})^2} \tag{4.2}$$

This utility represents the combined distance between the radio and other radios and the radio's distance from its channel of preference. As the radio gets closer to other radios and closer to its channel of preference, its utility improves. This time, each will receive different utilities as they balance preference and system goals. For $m = 4$, $c_{1*} = 1$ and $c_{2*} = 4$, the utility is shown in Table 4.3.

Table 4.3: Blind Rendezvous Proximity and Preference Coordination Game

		P2			
		Ch 1	Ch 2	Ch 3	Ch 4
P1	Ch 1	$(\frac{20}{20}, \frac{11}{20})$	$(\frac{15}{20}, \frac{7}{20})$	$(\frac{12}{20}, 7)$	$(\frac{11}{20}, \frac{11}{20})$
	Ch 2	$(\frac{10}{20}, \frac{6}{20})$	$(\frac{15}{20}, \frac{12}{20})$	$(\frac{10}{20}, \frac{7}{20})$	$(\frac{7}{20}, \frac{12}{20})$
	Ch 3	$(\frac{4}{20}, \frac{3}{20})$	$(\frac{7}{20}, \frac{10}{20})$	$(\frac{12}{20}, \frac{15}{20})$	$(\frac{7}{20}, \frac{15}{20})$
	Ch 4	$(\frac{2}{20}, \frac{2}{20})$	$(\frac{3}{20}, \frac{4}{20})$	$(\frac{6}{20}, \frac{10}{20})$	$(\frac{11}{20}, \frac{20}{20})$

Of particular interest in this table are the Nash Equilibria. Unlike the previous scenario, there now exists a NE which is not a rendezvous point and is not PO $(a_1 = 1, a_2 = 4)$, which exists at the point in which both radios are on the channel they prefer. In order to destabilize this NE, a time factor could be incorporated that decreases the radio's utility after repeatedly failing to rendezvous. The other

alternative would be to put heavier emphasis in the rendezvous utility factor than the channel preference utility factor.

Naturally, this model suffers from many of the same physical limitations as the proximity game. Coupled with the physical limitations of sensing, the lack of commonly named spectrum under the modular clock or free-for-all models would cause problems with analyzing channel preferences. It would also be possible to construct preference tables that would steer each radio towards non-common spectrum.

Symmetric Rendezvous Pursuit-evasion Game

As mentioned in Chapter 2, Lim [12] models an asymmetric rendezvous-evasion game with two evading agents R_1, R_2 and a searcher S. Although asymmetric rendezvous can produce more optimal solutions to this problem for the evaders, the ability to facilitate the roles may not exist in an ad-hoc cognitive network. Therefore, we examine rendezvous-evasion game in which the two rendezvousing agents must employ the same strategy. For this game, we abandon Lim's notation of using n as the number of locations in favor of m.

For our game theoretic analysis, we use the Anderson model from Chapter 3 and the symmetric algorithm proposed by Anderson and Weber in [2] as the strategy. For the ease of exposition, we assign a label to each of the m locations. The label is unknown to the agents or the searcher, and R_1, R_2, and S begin in different locations. By Anderson and Weber's optimal algorithm parameters, with a $\frac{1}{4}$ probability, the agent will choose to remain in its current location for m time steps. With $\frac{3}{4}$ probability the agent will choose a randomized permutation of the list of locations to visit in order. Lim alternatively defines this process as keeping track of all the locations visited and visiting only unvisited locations until all locations have been visited. Similar to the agents, for S we consider the strategy of waiting in location for m time steps and the strategy of visiting every location using a randomized permutation.

With $\frac{1}{16}$ probability, both agents will choose to remain in their current locations for m time steps. If S also chooses to wait, no rendezvous will exist by the mth time step and the game will not end. However, if S chooses to use the randomized permutation, S has a guaranteed utility of 1, i.e., he will jam one of them at some point.

With $\frac{6}{16}$ probability, exactly one of the agents will remain in their current location while the other agent searches exhaustively. In this scenario, we can use the analysis provided by Lim for his asymmetric rendezvous-evasion game. In doing this, the expected utility for S is bounded by $1 - e^{-1} = 0.632121$ as the number of locations m gets sufficiently large. If S chooses to wait instead, S will never find the stationary agent. If the game has not ended, the R_1 knows that S and R_2 are in separate locations, and that R_1 has not visited either of their locations so far in the permutation. Since there is no common labeling of locations between the players, at every step R_1 is as equally likely to find S as R_2. Therefore, the expected utility for S if S waits is 0.5, which makes searching the more optimal strategy.

Finally, with $\frac{9}{16}$ probability, both radios will be searching with a randomized permutation. By the analysis from Anderson and Weber, we denote the probability that the permutations of length m meet in k places as $prob(m, k)$. If two permutations meet in k places, Anderson also shows that the expected number of steps until the first meeting occurs is $\frac{m+1}{k+1}$. In order for R_1 and R_2 to achieve rendezvous, $k > 0$ and therefore our expected time of meeting is:

$$\sum_{k=1}^{m} prob(m, k) \cdot \frac{m+1}{k+1} \tag{4.3}$$

If S chooses to wait in place for m time steps while R_1 and R_2 search, we know that rendezvous with R_1 and R_2 is guaranteed by m time steps. Therefore $prob(m, 1) = 1$ between S and either R_1 or R_2 and $prob(m, k) = 0, k > 1$ because they can only rendezvous once. The expected time of S meeting R_1 and R_2 is therefore $1 \cdot \frac{m+1}{1+1} = \frac{m+1}{2}$.

Anderson gives us the following two equations for the expected time of meeting (E[TTR]) between the two searching agents R_1 and R_2

$$E[TTR] = \sum_{k=1}^{m} \left(prob(m,k) \cdot \frac{m+1}{k+1} \right) \tag{4.4}$$

$$E[TTR] = \left(\frac{m}{m-1} \right) \cdot ((m+1) \cdot (1 - prob((m+1),0) - prob(m,0) - 1)) \tag{4.5}$$

If the randomized permutations for R_1 and R_2 share no values in the same position, they are a *derangement* of each other. Using the well-known result of derangements [22], the probability that the sequences are a derangement is:

$$\lim_{m \to \infty} prob(m,0) = \frac{1}{e} \tag{4.6}$$

Using the fact that $prob(m,0) = \frac{1}{e}$, for large m, $(m+1) \approx m$, $prob((m+1),0) = \frac{1}{e}$.

$$E[TTR] = \left(\frac{m}{m-1} \right) \cdot \left((m+1) \cdot (1 - \frac{1}{e}) - \frac{1}{e} - 1 \right) \tag{4.7}$$

After reduction, we see that

$$E[TTR] = \frac{m(m \cdot e - m - 2)}{e(m-1)} \tag{4.8}$$

Therefore, the expected time for R_1 and R_2 to meet is $E[TTR] \approx m$.

Therefore, if S waits for m time steps the expected time of S meeting R_1 or R_2 is $\frac{m+1}{2}$ and the expected time of R_1 and R_2 meeting is approximately m. If S chooses to search while both R_1 and R_2 are searching, the expected time for S to meet R_1 and the expected time for S to meet R_2 would be m from the previous analysis. Since S has a lower expected time to meet one of the evading agents when it remains still, S should choose to wait.

With $\frac{7}{16}$ probability, the optimal strategy for S is to search using a randomized permutation of the m locations. With $\frac{9}{16}$ probability, the optimal strategy for S is to

remain in place for m locations. Therefore, given that R_1 and R_2 employ the mixed strategy of waiting with $\frac{1}{4}$ probability and searching with $\frac{3}{4}$ probability, the optimal mixed strategy for S is to wait with probability $\frac{9}{16}$ and search with probability $\frac{7}{16}$. [1]

In summary, the symmetric rendezvous pursuit-evasion game presents us with a minor extension of the asymmetric rendezvous pursuit-evasion game. The game provides some further insight into the difficulties faced when asymmetric strategies cannot be facilitated*. However, this game makes no claim to finding the best mixed strategy for R_1 and R_2 to minimize the expected utility for S. Also, the fact that the game ends when two or more players choose the same location limits the game's usefulness. A game which models this interaction without the "capturing" effect would be more useful to the field of cognitive radios and cognitive networks.

Modular Clock Blind Rendezvous Algorithm

The modular clock algorithm is a new rendezvous algorithm, designed to take advantage of some of the guarantees in number theory often leveraged by cryptography. By using prime number modulation, the algorithm is able to use some of these guarantees inherent to the CRT and the modular inverse.

Notation. The notation is as follows for a single radio:

- r is the rate that the cognitive radio hops channels. Every time slot, the radio hops forward this many channels in its set, and wraps around when it reaches a channel greater than the maximum number (modulo arithmetic).

- t is the time slot of the system.

- τ is the starting channel index at some arbitrarily chosen common time reference.

Modular Clock Algorithm. Algorithm 1 outlines the pseudocode for the modular clock (MC) algorithm. In order to leverage the power of prime numbers, we

---

[1]Note that this combination of strategy is has not been analyzed in terms of being a NE or being PO

must find a prime number p to use for our algorithm. For MC, we choose the lowest prime greater than the number of channels m. Since there will be a gap between the actual number of channels available and the number of channels that the radio could attempt to use (i.e. between m and p), for all channel indices that the clock selects between m and p, we tell the radio to choose a channel index randomly between 1 and m.

Note that all variables listed in Algorithm 1 are relative to the radio which is executing the algorithm, and not global values. Also, the channel offset j is relative to the first channel in its set of channels c_1. The channel that the radio actually chooses is c. For m channels, the channel indices range from 0 to $m-1$.

Algorithm 1 Modular Clock Algorithm (MC)

1: observe m, the number of channels available
2: calculate p, the next largest prime to m
3: current channel $c = c_\tau$
4: current offset $j = \tau$
5: **while** not rendezvous **do**
6:     choose r from 1 to $p-1$ randomly
7:     **for** $t = 0$ to $2p$ **do**
8:         $j = (j + r) \mod (p)$
9:         **if** $j < m$ **then**
10:            $c = c_j$
11:         **else**
12:            $c = c_{rand([0,m-1])}$
13:         **end if**
14:     **end for**
15: **end while**

We begin the algorithm with our channel index $j = \tau$. For every time step, we increment our index j by r, then modulate j by p. If the resulting j is between 0 and $m-1$, then the radio moves to channel c_j. If the resulting j is greater than $m-1$, then the radio chooses a random index between 0 and $m-1$ and moves to that channel. Note that j is not changed when $j > (m-1)$. If we have not achieved rendezvous after $2p$ time steps, we randomly choose a new r value between 1 and $p-1$. The algorithm terminates at any point in which rendezvous is achieved.

Algorithm Analysis. Now that we know the means by which each radio will attempt to achieve rendezvous, we analyze the interaction between the radios and the sequences they execute. The channel index that radio i is on at a particular time step t can be given as:

$$j = t \cdot r_i + \tau_i \mod (p_i), p_i \geq m_i \tag{4.9}$$

It's important to note that the term "convergence" refers to arriving to a common channel index. Depending on the model used and the amount of channels available, this may or may not correspond to rendezvous.

Theorem 1. *Under the DaSilva model (where $C_i = C, \forall i \in N$, $m_i = m_j$, and $p_i = p_j$ since all radios perceive the same set of open channels), when $r_i \neq r_j$, convergence occurs in $t < p$ steps once both radios start the rendezvous process.*

Proof. Define $\delta_r = r_j - r_i$, $\delta_\tau = \tau_j - \tau_i$

Rendezvous occurs when $t \cdot r_j + \tau_j \mod (p) \equiv t \cdot r_i + \tau_i \mod (p)$. We can simplify the rendezvous equation to:

1. $t \cdot \delta_r \equiv -\delta_\tau \mod (p)$

2. The modular multiplicative inverse of any number $a \mod (p)$ exists iff a and p are co-prime. That is, the Greatest Common Divisor (GCD) of a and p is 1.

3. Since $r_i \neq r_j$, $\delta_r \neq 0$

4. Since $1 < r_i < p$ and $1 < r_i < p$, $-p < \delta_r < p$.

5. Since $-p < \delta_r < p$ and p is prime, $gcd(\delta_r, p) = 1$

6. Therefore there exists a modular multiplicative inverse of δ_r, δ_r^{-1}

7. $t \cdot \delta_r \cdot \delta_r^{-1} \equiv -\delta_\tau \cdot \delta_r^{-1} \mod (p)$

8. $t \equiv -\delta_\tau \cdot \delta_r^{-1} \mod (p)$

9. Since $-\delta_\tau \cdot \delta_r^{-1} \mod (p)$ is a value between 0 and $p - 1$, our t value is between 0 and $p - 1$ and we have proven rendezvous will occur in $t \leq p$ steps.

$\square$

Radio Timing. By waiting until $2p$ time steps have occurred before timing out and switching r values, the radios can ensure that they have had at least p time without either radio changing r values. For example, if radio 1 begins a particular r sequence at $t = 0$, while radio 2 begins theirs at $t = p - \epsilon$. If radio 1 waits until $t = 2p$ to change r values, then they will have maintained static r values for $2p - (p - \epsilon)$ or $p + \epsilon$ time together, which guarantees rendezvous. If radio 2 begins their sequence at $t = p + \epsilon$ time, we can perform the same analysis by calling $p + \epsilon$ time 0, which means radio 1 will reset its r value at $2p - p + \epsilon$ time, or $t = p - \epsilon$, and the previous analysis applies.
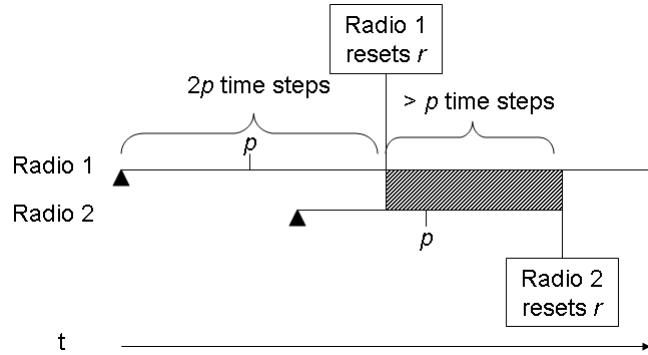

Figure 4.4: Using $2p$ reset timeout to guarantee p common time slots before reset

Algorithm Analysis With Same Rate.

Theorem 2. *Under the DaSilva model (where $\forall C_i, C_i = C$, $m_i = m_j$, and $p_i = p_j$ since all radios perceive the same set of open channels), when $r_i = r_j$, and $\tau_i \neq \tau_j$, rendezvous will not occur.*

Proof. Recall that $r_j - r_i = \delta_r$, $\tau_j - \tau_i = \delta_\tau$

Rendezvous occurs when $t \cdot r_j + \tau_j \mod (p) \equiv t \cdot r_i + \tau_i \mod (p)$.

Proof by contradiction:

1. Assume that $t \cdot \delta_r \equiv -\delta_\tau \mod (p)$

2. Since $0 \leq \tau_i < p$ and $0 \leq \tau_j < p$, $-p < \delta_\tau < p$

3. Also, since $r_j = r_i$, $\delta_r = 0$

4. Therefore, $t \cdot 0 \equiv -\delta_\tau \mod (p)$

5. This reduces to $0 \equiv -\delta_\tau \mod (p)$

6. By our assumptions, $\tau_j \neq \tau_i$, $\delta_\tau \neq 0$

7. Since $-p < \delta_\tau < p$ and $\delta_\tau \neq 0$, $0 \neq -\delta_\tau \mod (p)$

Therefore, we have a contradiction, and $t \cdot \delta_r \neq -\delta_\tau \mod (p)$.

$\square$

Since the δ_r value is 0, the sequences maintain the same difference in channel value (δ_τ) they began with. However, rendezvous may still occur when the sequence chooses an index between m and p and the radio chooses a random index from 0 to $m - 1$. The expected TTR is rather high in this case, so we introduce a timeout feature at $2p$.

By Theorem 1, if $r_i \neq r_j$ convergence is guaranteed within p time steps. There-fore, the radios can observe that they have failed to rendezvous within $2p$ time and change their r values. This is what line 7 is doing in Algorithm 1. Note that with a $\frac{p-1}{p}$ probability, they will select different r values and Theorem 1 will apply. However, with $\frac{1}{p}$ probability, they will select the same r value again. In case where we have hit our $\frac{1}{p}$ chance of an orthogonal sequence, the radio will observe failure to rendezvous after $2p$ time steps. If we fail to rendezvous by the pth change of rate r, we will have spent $2p \cdot p$ time. The odds of failing to rendezvous for every $2p$ iteration is $\frac{1}{p}$. The probability of failing to rendezvous after p changes of r values is $\frac{1}{p} \cdot \frac{1}{p} \cdot \frac{1}{p} \ldots = \frac{1}{p^p}$. Therefore, with $1 - \frac{1}{p^p}$ probability, we will do no worse than $O(p^2)$ time.

Since we know that with $\frac{p-1}{p}$ probability we will choose different r values and rendezvous within $2p$ time, and with $\frac{1}{p} \cdot \frac{p-1}{p}$ probability we will rendezvous in $2 \cdot (2p)$ time, we can use the following infinite series to find an upper bound on expected TTR

$$E[TTR] \leq (2p \cdot \frac{p-1}{p}) + (4p \cdot \frac{1}{p} \cdot \frac{p-1}{p}) + (6p \cdot \frac{1}{p^2} \cdot \frac{p-1}{p}) + ... \qquad (4.10)$$

This series reduces to the closed form expression:

$$E[TTR] \leq 2p + \frac{2p}{p-1} \qquad (4.11)$$

Therefore, our upper bound on TTR is only slightly larger than $2p$.

Theorem 3. *Under the modular clock model, ($C_i \neq C_j$, $\bar{C} = C_i \bigcap C_j \neq \emptyset$) and $p_i \neq p_j$, MC rendezvous will occur within $p_i \cdot p_j$ time steps.*

Call the larger of the two p values p_i and the smaller p_j. The radio which uses p_i will be referred to as r_i and use τ_i. Note that p_i and p_j approximate m_i and m_j. To prove the Rendezvous exists, we need to be able to find out when $t \cdot r_i + \tau_i$ mod $(p_i) = t \cdot r_j + \tau_j$ mod (p_j).

Proof. 1. Recall from the CRT, if $\gcd(p_1, p_2) = 1$, then for any a_1 and a_2, there exists a solution for

$$x \equiv a_1 \mod (p_1)$$
$$x \equiv a_2 \mod (p_2)$$

2. Since p_i and p_j are prime and not equal, they are also co-prime

3. Since p_i and p_j are co-prime, $\gcd(p_i, p_j) = 1$

4. Substitute $t \cdot r_i + \tau_i$ for a_1 and $t \cdot r_j + \tau_j$ for a_2

$$x \equiv t \cdot r_i + \tau_i \mod (p_i)$$
$$x \equiv t \cdot r_j + \tau_j \mod (p_j)$$

5. By the CRT, there exists an x which solves that set of equations, and rendezvous is guaranteed.

52

□

However, we can say even more about how frequently this rendezvous will occur. If $\bar{m} = m_j$, then within $p_i \cdot p_j$ time steps, rendezvous will occur m_j times. We know this because we can set x equal to any value in the range $\{0, 1, 2, ...(m_j - 1)\}$ and solve the set of equations for a unique $t < (p_i \cdot p_j)$. Since we have m_j rendezvous points over the span of $p_i \cdot p_j$ time and $p_j \approx m_j$, our $E[TTR] \approx p_i$.

We can extend this analysis to a more general conclusion about the expected TTR. In the scenario provided above, we created the situation where two radios shared an overlap of p_j common rendezvous channels, m_j of which translated to non-virtual channel indexes. In a more general sense, the approximate TTR can be re-written as $\frac{\bar{m}}{m}$. As long as $\bar{C} \neq \emptyset$, rendezvous will occur in at most $\frac{p_i p_j}{1}$ time steps (since p_i and p_j are both factors of their m values), which is asymptotical to $\theta(m^2)$ time.

Degenerative Case. The following is the analysis of the degenerative case for the modular clock algorithm. Under the modular clock model, where $\exists C_i \neq C$ and not all radios share the same perception of the availability, assume that $C_i \neq C_j$, $C_i \bigcap C_j \neq \emptyset$, and $m_i = m_j$. In other words, two radios have two different sets of channels with some overlap but the same quantity of observed channels.

Since $m_i = m_j$, $p_i = p_j$. Under this case, the CRT cannot apply because the moduli are factors of each other(not co-prime). Similarly, the analysis under Theorem 1 cannot apply because the convergence point is not guaranteed to be in $C_i \bigcap C_j$. Through simulation it has been observed that convergence may still occur under these assumptions, but no guarantee is possible.

Random Strategy vs. Modular Clock Algorithm. Anderson and Weber proved that the optimal rendezvous solution for $m = 2$ is random under the Anderson model, although they were unable to prove it for anything larger. We also know that if we use the modular clock algorithm under the DaSilva model and $r_i \neq r_j$, convergence is guaranteed in fewer than p time steps. As noted in Chapter 3, the DaSilva model

and the Anderson model are very similar. Therefore, there may be crossover point at which the MC algorithm has a lower expected TTR than random.

Recall that if $r_i = r_j$, we are not likely to rendezvous and must wait $2p$ time steps to reset r values. If this is not the case, we are guaranteed convergence in p time steps. The probability that $r_i = r_j$ is equal to $\frac{1}{m}$, and the probability that $r_i \neq r_j$ is $1 - \frac{1}{m}$. Also recall that the expected TTR for the random approach is m.

To find the crossover point, we plot the probability that we can guarantee rendezvous ($r_i \neq r_j$) under the MC algorithm against the probability of achieving rendezvous in fewer than $2p$ time steps using a random approach. The probability that random achieves rendezvous by $2p$ time steps is $1 - (1 - \frac{1}{p})^{2p}$. In Figure 4.5, we plot the two against each other and see that based on this model for $m < 9$, random should perform better than the modular clock algorithm for this particular metric. Therefore under the DaSilva model, if the radio observes fewer than nine channels, it should implement a random search instead of the modular clock algorithm.

Modified Modular Clock Algorithm

To avoid the degenerate case in the MC algorithm where $p_i = p_j$ but neither $C_i \subseteq C_j$ nor $C_j \supseteq C_i$, we randomize our primes within a certain range to avoid having the same prime. This modified modular clock (MMC) algorithm is shown in Algorithm 2. Under the MMC algorithm, we select a random prime number p from the set of primes between m and $2m$.

Again, for all action spaces between m and our chosen p, we tell the radio to choose randomly. Since p is no larger than $2m$, we will spend no more than half of our time choosing randomly. If we can avoid having the same prime, we are guaranteed solutions for rendezvous within $p_i \cdot p_j$ time by Theorem 3. Since our main concern at this point is avoiding the same prime number, one strategy is to change our prime numbers if we have failed to notice rendezvous in $2p^2$ time. This is not guaranteed to
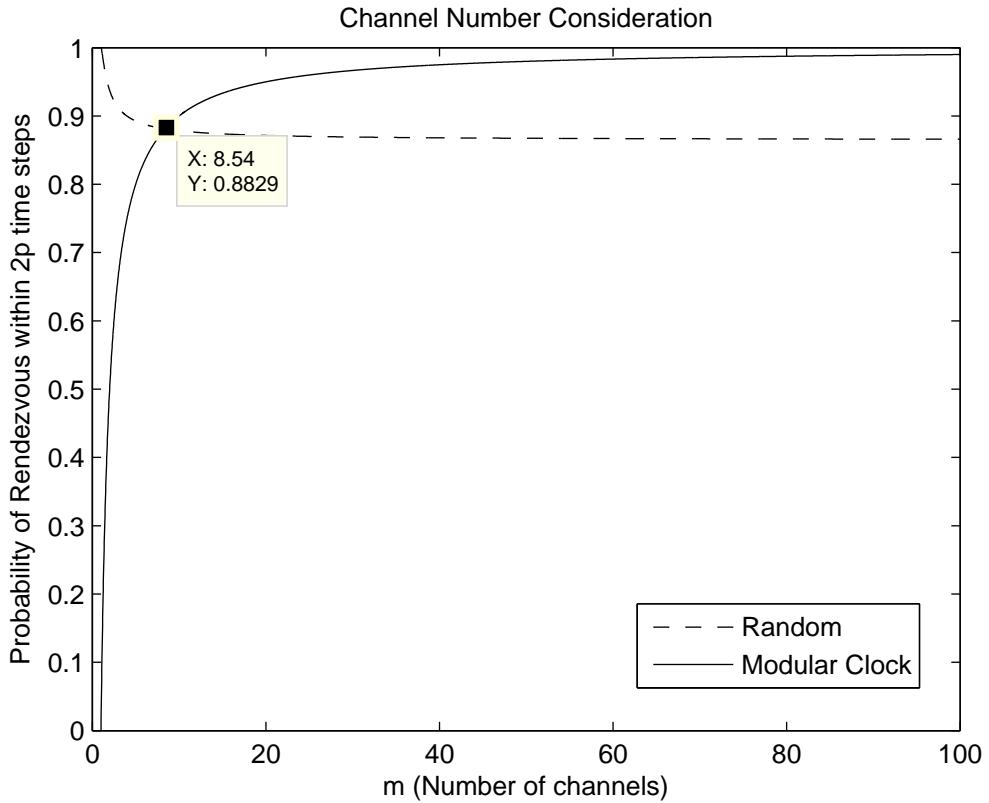
Figure 4.5: Probability of rendezvous in fewer than $2p$ slots

be the correct timeout. If there exists a large difference between p_i and p_j, the radio with the smaller p value may reset too quickly.

By the Prime Number Theorem, the number of primes from 1 to x can be approximated by $\frac{x}{\ln(x)}$. Therefore, the number of primes between m and $2m$ is approximately $\frac{2m}{\ln(2m)} - \frac{m}{\ln(m)}$. The goal of the MMC algorithm is to avoid choosing the same prime numbers (i.e. avoid $p_i = p_j$). Figure 4.6 evaluates the probability of choosing the same prime number (p_i, p_j) based on the number of channels for each radio (m_i, m_j). For example, since both radios choose their prime between m and $2m$, if $m_j \leq \frac{1}{2} \cdot m_i$, the probability of choosing the same prime is 0. Consequently, if $m_j = m_i$, then the probability of choosing the same prime is $\frac{2m_i}{\ln(2m_i)} - \frac{m_i}{\ln(m_i)}$.

Figure 4.6 shows that it requires 10% of the channels to have a 10% probability of choosing the same prime if $m_j = 0.55 \cdot m_i$ than if $m_j = m_i$.

55

Algorithm 2 Modified Modular Clock Algorithm (MMC)

1: observe m, the number of channels available
2: current channel $c = c_\tau$
3: current offset $j = \tau$
4: **while** not rendezvous **do**
5:    choose r from 1 to $m-1$
6:    choose p, a prime between m and $2m$
7:    **for** $t = 0$ to $2p^2$ **do**
8:       $j = (j+r) \mod (p)$
9:       **if** $j < m$ **then**
10:          $c = c_j$
11:      **else**
12:          $c = c_{rand([0,m-1])}$
13:   **end if**
14:    **end for**
15: **end while**
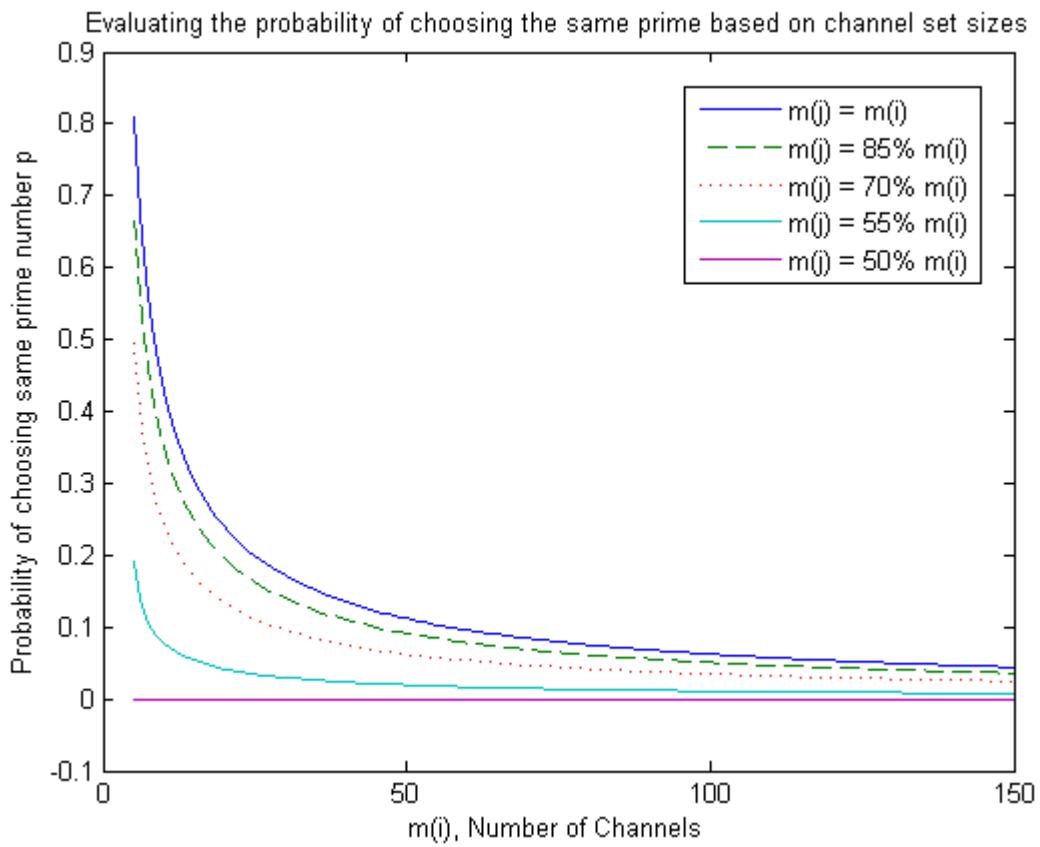
Figure 4.6: Evaluating the probability of choosing the same prime number

Modular Clock and a Wait Factor

In Anderson and Weber's paper [2], they propose the use of a wait factor θ. θ represents the probability in which either radio will remain in its channel and wait for the other radio to come find it for a prescribed period. For our purposes, θ would be the probability in which the radio overrides its random step size selection of r in the MC algorithm and sets $r = 0$. Assuming the radios are synchronized and decide their wait factors θ at the same time, the radio would wait m time steps before finding a new r.

In the paper, Anderson and Weber used random permutations of the m channels and the probability of derangement to consider whether the sequences would converge at any point. Using this approach, the optimal wait factor θ was calculated to approach 0.2475 as the number of channels m gets infinitely large.

Using a similar approach, we created a formulation to evaluate a wait period for the original modular clock algorithm. Using the Anderson and Weber model, we assume $C_i = C_j$. We also assume that the radios are synchronized and slotted. The DaSilva model was not used due to the inability to assume synchronization. By the MC algorithm, both radios choose the same p, and they choose the same rate r with $\frac{1}{m}$ probability. However, we assume that $r \neq 0$, so there are only $m - 1$ different r values to choose. Therefore, the radios choose the same r with $\frac{1}{m-1}$ probability, and choose different r with $1 - \frac{1}{m-1} = \frac{m-2}{m-1}$ probability.

- Case 1 – Both radios wait, neither finds the other in m time and must start over. This occurs with θ^2 probability.

- Case 2 – One radio waits and the other searches. $E[TTR] = \frac{m}{2}$. This occurs with $2\theta - \theta^2$ probability.

- Case 3 – Both radios search, and choose different rates, similar to Theorem 1. $E[TTR] = \frac{m}{2}$.

- Case 4 – Both radios search, and choose the same rate, similar to Theorem 2. Radios wait m time and must start over.

Since we have two cases in which we fail to rendezvous, the equation attempts to solve for the optimal amount of time in which we favor Case 1 vs. Case 4. We must find the optimal value of θ that minimizes t for any given number of channels m.

$$t = \theta^2(m+t) + (2\theta - \theta^2)\left(\frac{m}{2}\right) + (1-\theta)^2\left(\frac{m}{2}\right)\left(\frac{m-2}{m-1}\right) + (1-\theta)^2(m+t)\left(\frac{1}{m-1}\right) \tag{4.12}$$

Solving for t, this equation reduces to the following:

$$t = \frac{2\theta^2 m^2 - \theta^2 m - 2\theta m + m^2}{2m - 2\theta^2 m + 4\theta - 4} \tag{4.13}$$

Plotting this equation as graph for a variety of m inputs, we see the optimal value of θ for $m > 2$ is zero, according to Figure 4.7. This can be explained by considering the fact that Anderson and Weber used randomized permutations of the channel set. These permutations have probabilities of rendezvous but no guarantees, in contrast to the MC algorithm. Anderson and Weber only had guaranteed rendezvous in the case where one player waited while the other searched. Using the modular clock algorithm where convergence is guaranteed when both players search using different rates r, we lose our incentive to risk the scenario in which both players wait.

Simulation and Results

Due to the slotted and discrete nature of the modular clock model, simulation of rendezvous algorithm can be performed in a number of environments such as MATLAB or Maple. The platform chosen for developing the simulation was Cygwin running on Windows Vista. The programs were written in C++ and compiled with g++ 3.4.4 with no flags. A copy of the source code used is provided in Appendix 1. Execution of experiments and collection of data was handled through a
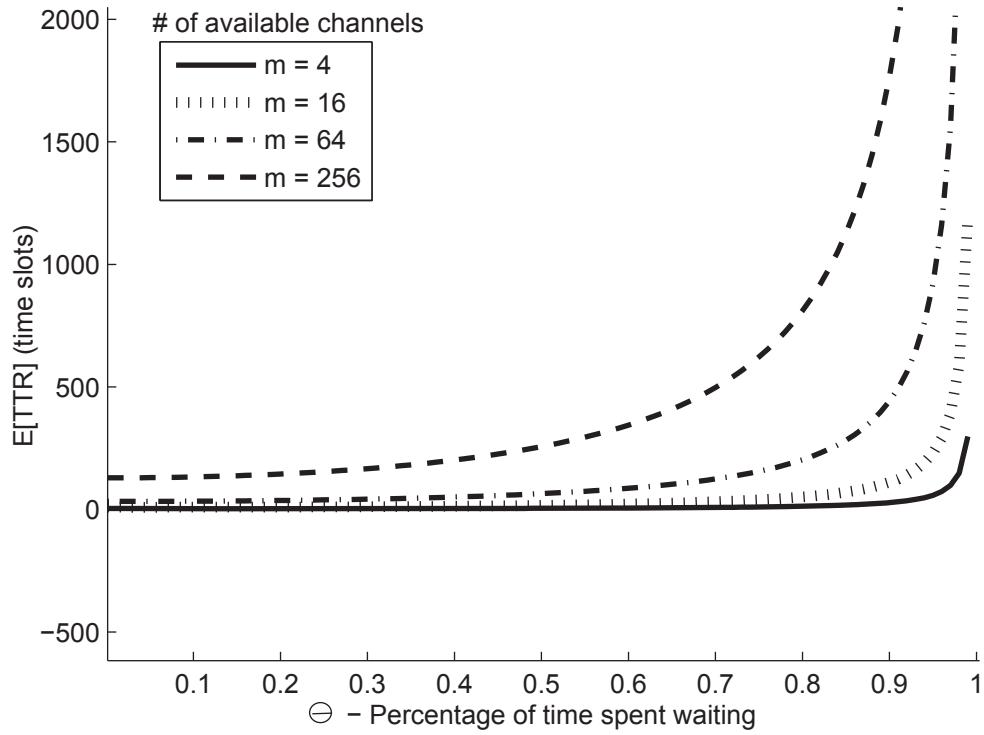
Figure 4.7: Effect of a Wait Factor

set of Bash shell scripts which parsed through the master output and collected the rendezvous solution points in a tab delimited file. Once the results were collected for the experiments, the results were imported into Minitab and analyzed.

Design of Experiments. In designing the experiments for the modular clock algorithm and the modified modular clock algorithm, a number of hypotheses were tested. In order to distinguish between the modular clock algorithm and the modified modular clock algorithm, the modular clock algorithm is typically referred to as "MC" and modified modular clock is referred to as "MMC." To evaluate these statements we evaluate the estimated TTR, or mean of the simulations. We'll also look at some other statistical information to evaluate performance, such as the value of percentiles.

Hypothesis 1. *When $m_i = m_j < 9$ and $C_i = C_j$, random performs better (lower average TTR) than both MC and MMC. When $m_i = m_j > 9$ and $C_i = C_j$, MC will perform better than random as m increases.*

The information presented in Figure 4.5 is the basis for these hypotheses. According to Figure 4.5, a random approach has a better chance of achieving rendezvous in fewer than $2p$ time slots if $m < 9$. Conversely, the information tells us that MC has a better chance of achieving rendezvous in fewer than $2p$ slots for $m > 9$.

Hypothesis 2. *For $m_1 > 9$ and $m_2 > 9$, if the prime value selected under MC by radio 1 is equal to the prime value selected by radio 2 ($p_1 = p_2$) and neither $C_1 \subseteq C_2$ nor $C_2 \subseteq C_1$, then MMC performs better than MC.*

While this scenario sounds far-fetched, it is actually fairly easy to construct. Given that two radios observe a large set of common channels and have approximately the same number of total channels, the same prime number will be chosen by MC.

Hypothesis 3. *When there exists a large difference between m_i and m_j, random should perform better than MC and MMC.*

Because both MC and MMC are symmetric and have a timeout window, this hypothesis highlights the case in which the timeout window would get abused. The radio with the smaller number of channels would trigger a timeout prematurely.

Hypothesis 4. *When $m_1 > 9$, $m_2 > 9$ and $m_1 \approx m_2$, as $\bar{C}$ gets smaller, MMC should perform better than random and MC.*

As the number of common channels decreases, MMC maintains a guarantee of rendezvous in $p_i \cdot p_j$ time. This guarantee should provide better results as the number of common channels decreases.

Trials.   The following is a quick listing of the parameters for the trials. All trials were conducted using two radios with a random starting channel.

- Trial 1 - between 3 and 18 channels, all channels in common

- Trial 2 - 25 channels, 25 in common

- Trial 3 - 50 channels, 50 in common

- Trial 4 - 25 and 20 channels, 20 in common

- Trial 5 - 25 and 5 channels, 5 in common

- Trial 6 - 25 channels each, 5 in common

- Trial 7 - 25 and 10 channels, 5 in common

- Trial 8 - 25 channels each, 1 in common

Trial 1. Trial 1 is designed to test Hypothesis 1. According to Hypothesis 1, random should perform best for $m < 9$. For the trial, we have m values between 3 and 18, and $\bar{C} = m$.
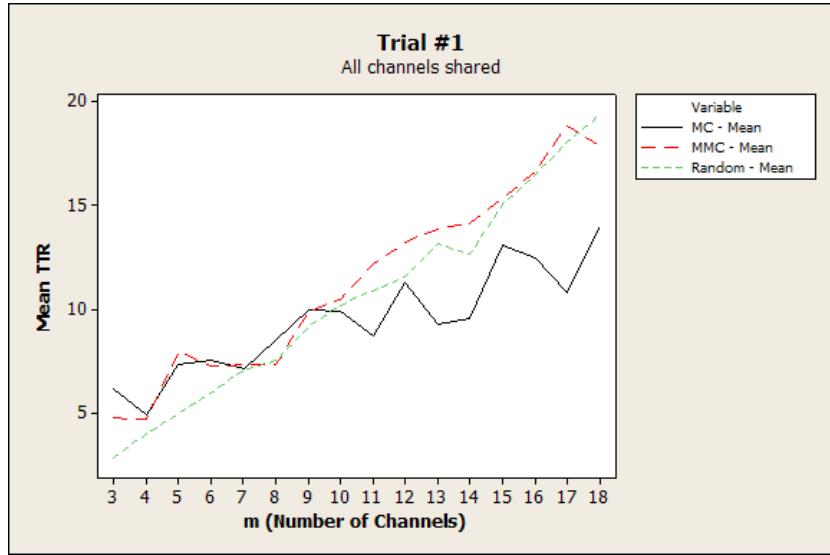

Figure 4.8: Mean TTR for m from 3 to 18, $\bar{C} = m$

Figure 4.8 shows the mean TTR for each algorithm. As expected, the mean TTR for random grows linearly with m. According to the figure, random performs better than MC until $m = 7$, then a little better than MC until $m > 10$, where MC performs better at every m value. Although the figure does not clearly show a crossover point, it does support the hypothesis that the crossover point is in the neighborhood of $m = 9$.

Whereas MC began to perform better than random by $m = 10$, the mean TTR for MMC was indistinguishable from random throughout most of the trial. Only at $m = 5$ are the two means significantly different, which suggests an aberration.

Trial 2.   Trial 2 is a further test of Hypothesis 1. For the trial, we have set $m_1 = m_2 = 25$, and $\bar{C} = 25$. Trial 2 is designed to continue to observe the behavior of the three algorithms as m begins to get larger.
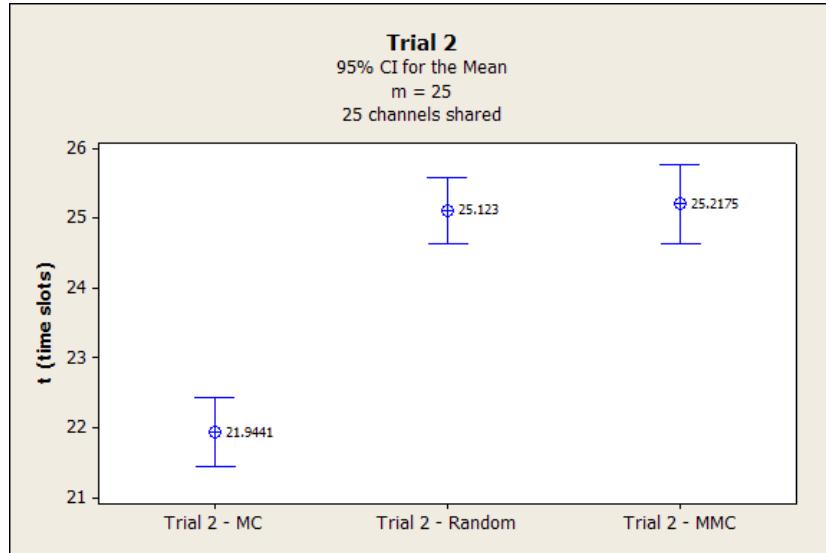

Figure 4.9: 95% mean confidence interval for $m_1 = 25$, $m_2 = 25$, $\bar{C} = 25$

In Figure 4.9 we see that the mean value of MC is easily statistically smaller than the mean of the random approach. In fact, the mean value of MC population is 13% smaller than the mean values of the other two approaches. As we get further from our crossover, the mean TTR for MC is becoming smaller than the mean TTR for random. The means of random and MMC continue remain statistically indistinguishable.

By Figure 4.10, we see that the median value for MC is lower than the median values of both random and MMC in Trial 2. Furthermore, the 25th percentile, 75th percentile, and maximum values of MC are lower than equivalent values for the other two approaches.
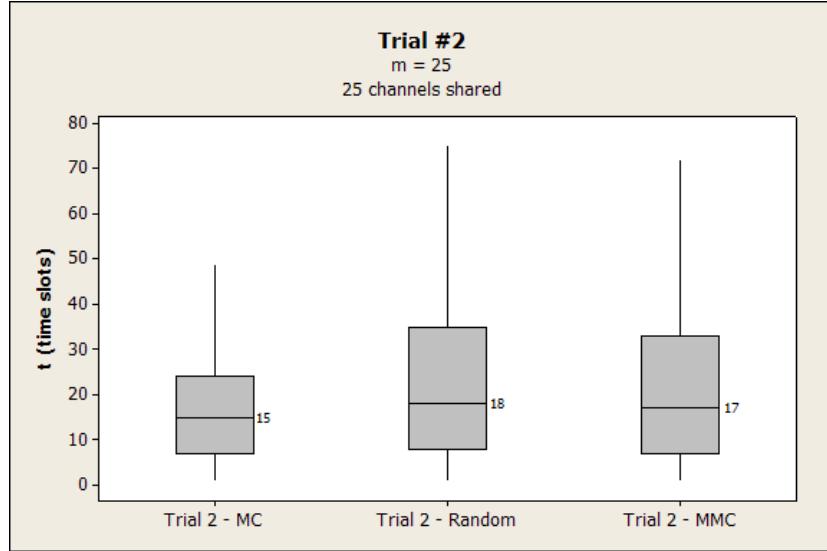
Figure 4.10: Box plot for $m_1 = 25$, $m_2 = 25$, $\bar{C} = 25$

Trial 3. Trial 3 was designed to continue to observe the behavior of the algorithms as m gets even larger. The behavior proposed by Hypothesis 1 and observed in Trial 2 should be continued and exaggerated in this trial.
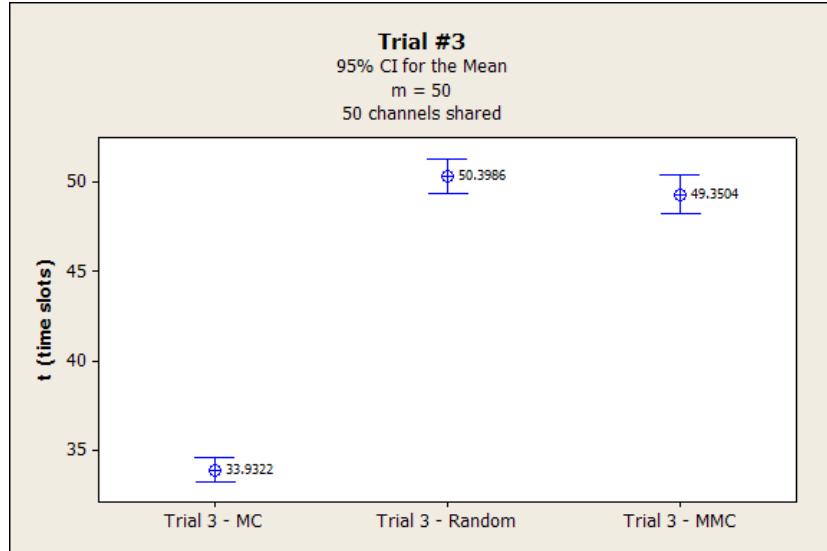

Figure 4.11: 95% mean confidence interval for $m_1 = 50$, $m_2 = 50$, $\bar{C} = 50$

Figure 4.11 shows that MC improves over random, as the mean for MC is nearly 33% lower than the random mean. As m gets larger, we expect the mean of MC to trend towards $\frac{m+1}{2}$, while the mean of random should remain at exactly m. If there

63

were no probability of choosing the same r values, the mean of MC would equal $\frac{m+1}{2}$. Instead, as m gets larger the probability of choosing the same r value decreases, which brings the mean closer to $\frac{m+1}{2}$.
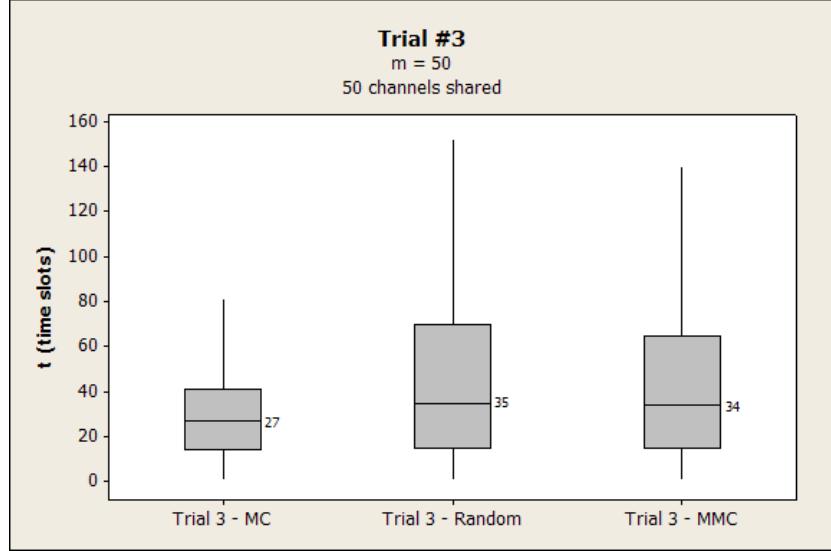

Figure 4.12: Box plot for $m_1 = 50$, $m_2 = 50$, $\bar{C} = 50$

Figure 4.12 further demonstrates the growing difference between MC and the random approach. According to the figure, the 75th percentile result from MC is about on par with the median of the random. Also, the maximum value of MC is half the maximum value of the random solution. As the probability of choosing the same rate decreases, the guarantees provided by the algorithm while under DaSilva model become the dominant factor.

Trial 4. Trial 4 demonstrates one of the simplest cases for MC when $m_i \neq m_j$. In this trial, $m_1 = 25$, $m_2 = 20$, $\bar{C} = 20$, and $C_2 \subset C_1$. In this trial, there exist 20 unique solutions which contain rendezvous out of $25 \cdot 20 = 500$ possible outcomes. Therefore, our analysis indicates E[TTR] is approximately 25.

Figure 4.13 confirms that our E[TTR] analysis can be accepted with 95% confidence for all three algorithms. We would also expect that the maximum value would be less for MC and MMC compared to random. As we can see in Figure 4.14 however,
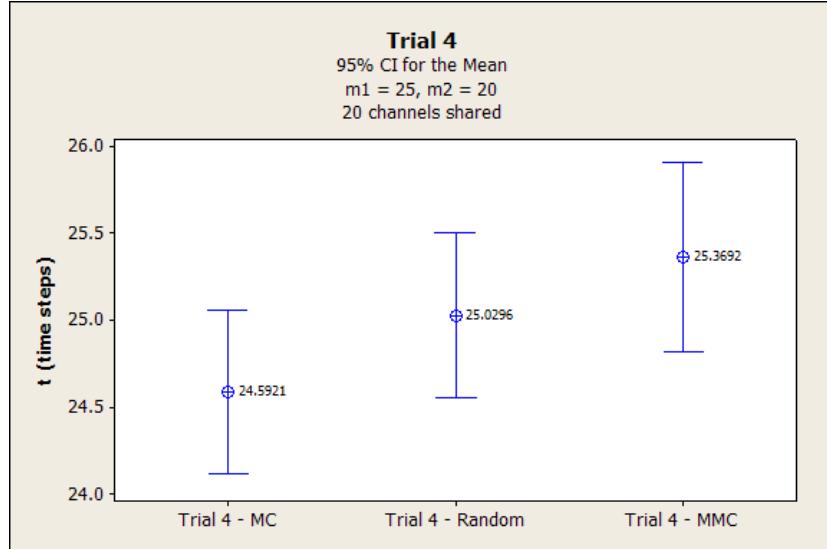
Figure 4.13: 95% mean confidence interval for $m_1 = 25$, $m_2 = 20$, $\bar{C} = 20$

this does not occur. As a base case for Hypothesis 4, we see that when $\bar{C}$ is large compared to m_1 and m_2, MMC does not outperform the other algorithms.
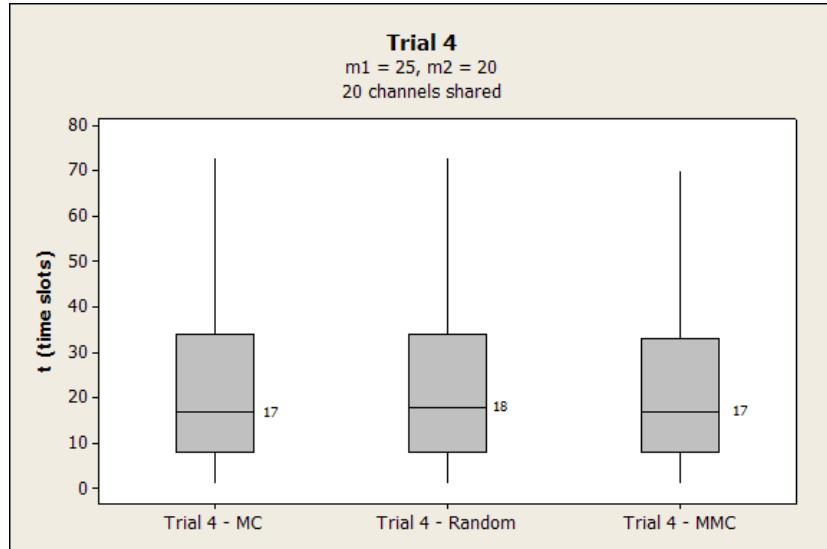
Figure 4.14: Box plot for $m_1 = 25$, $m_2 = 20$, $\bar{C} = 20$

Trial 5. Trial 5 is intended to stress the timeout function of MC and MMC and test Hypothesis 3. In this trial, we set $m_1 = (m_2)^2$. This case is problematic for the CRT due to the fact that each radio times out and changes rate values based

on their local p values ($2p$ steps for MC, $2p^2$ steps for MMC). When the number of channels differs greatly, the radio with a smaller number of channels will reset its sequence quickly compared to the radio with a larger number of channels. By the analysis performed in Theorem 2, the radios must remain in their sequence for $p_l \cdot p_s$ time steps to guarantee rendezvous.
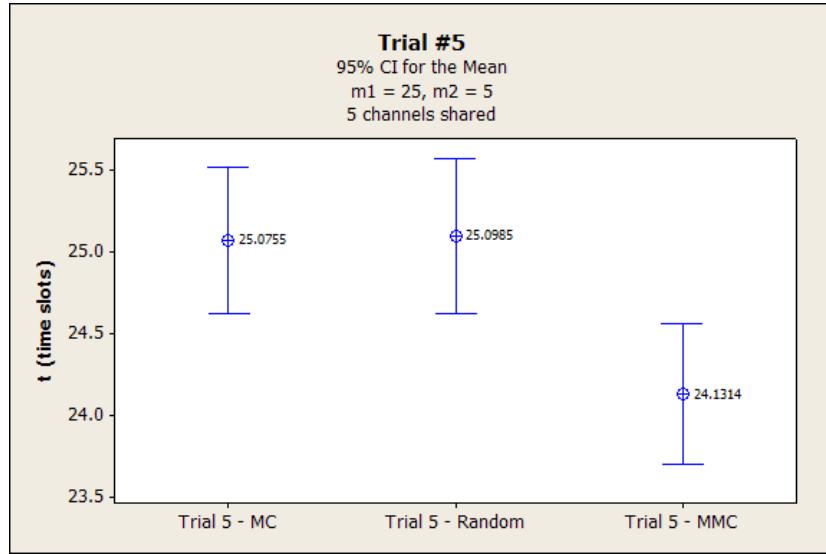

Figure 4.15: 95% mean confidence interval for $m_1 = 25$, $m_2 = 5$, $\bar{C} = 5$

Figure 4.15 strongly refutes Hypothesis 3. In particular, Figure 4.15 demonstrates that the mean value of MMC is statistically smaller than the mean value of the other two algorithms with 95% confidence. The success of MMC in this trial can be explained by the algorithm's use of prime numbers between m and $2m$, which provides a larger set of p values to choose from.

As we can see from Figure 4.16, the median results of all three algorithms is approximately equivalent. The percentile values for MC and the random approach are nearly equal at every level. However, the 75th percentile and 100th percentile values for MMC are quite a bit lower than the previous two.

66
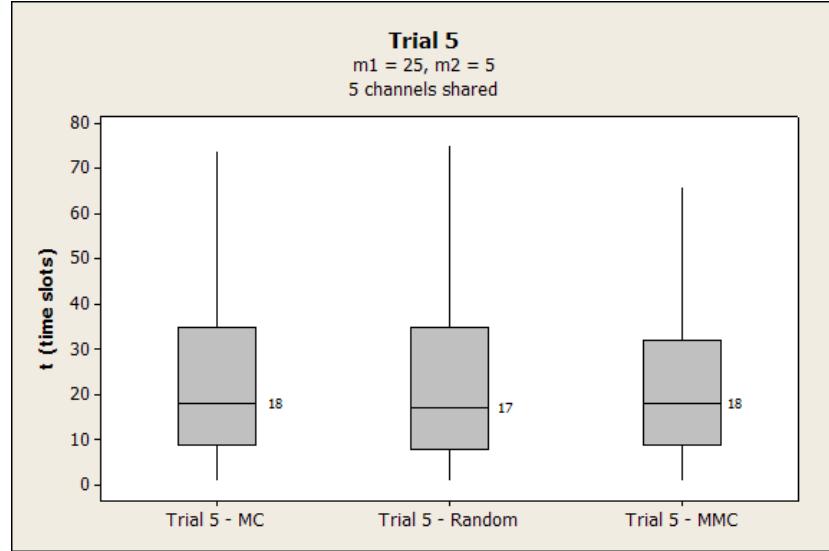
Figure 4.16: Box plot for $m_1 = 25$, $m_2 = 5$, $\bar{C} = 5$

Trial 6.   Trial 6 is designed to test Hypothesis 2. It is also designed to further analyze the behavior of the algorithms under the scenario discussed in Hypothesis 4. In this trial, $m_1 = m_2$ and $|\bar{C}| = \sqrt{m_1}$.
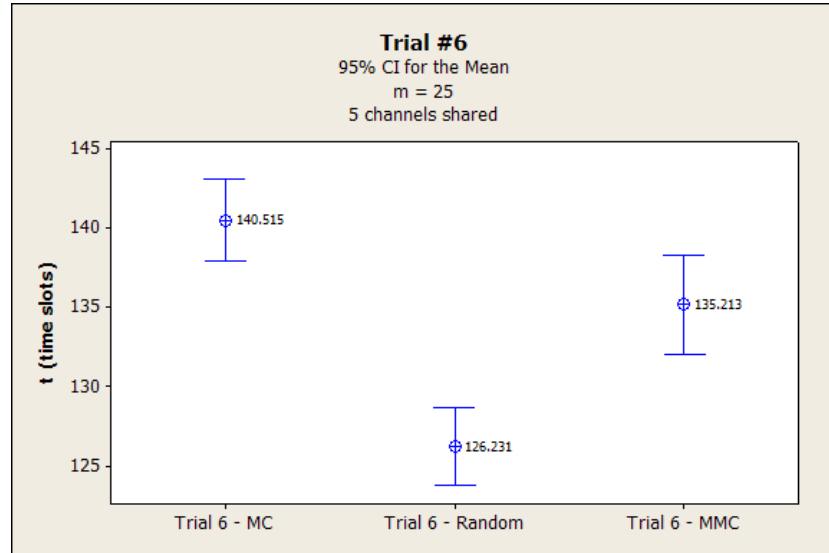
Figure 4.17: 95% mean confidence interval for $m_1 = 25$, $m_2 = 25$, $\bar{C} = 5$

According to Figure 4.17, the mean value of the random algorithm is statistically smaller than the mean value of MMC with 95% confidence. Trial 6 seems to refute, rather than support Hypothesis 4.
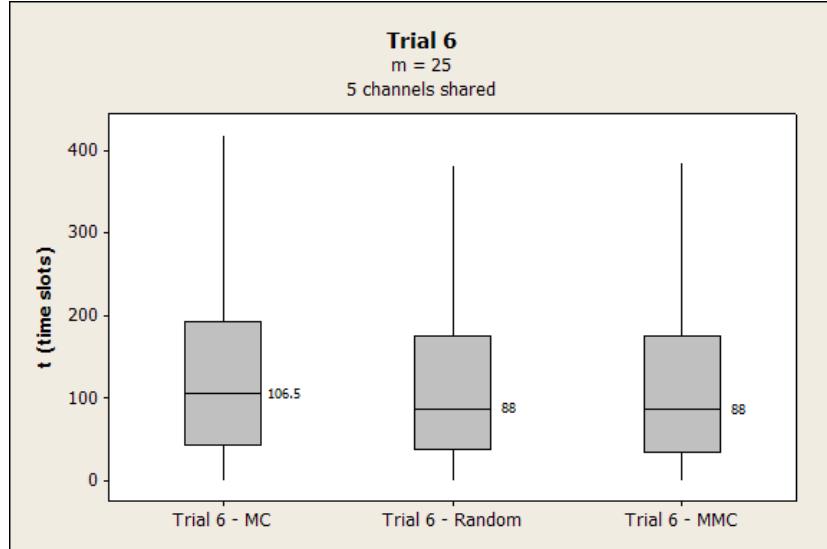
Figure 4.18: Box plot for $m_1 = 25$, $m_2 = 25$, $\bar{C} = 5$

Figure 4.18 tells us that MC is beginning to perform rather poorly under this trial. This behavior is expected under the analysis provided in the MC's degenerative case. According to the box plot, MMC performs almost exactly like random for this trial.

Trial 7.   Trial 7 is designed to examine the behavior of MC and MMC under a combination of Trials 4, 5, and 6. For this trial, we set $m_1 = 25$, $m_2 = 10$, and $\bar{C} = 5$.

Figure 4.19 confirms that the mean values of MC and MMC are statistically smaller than the mean of the random algorithm. The figure also demonstrates that the mean value of MMC in the trial was 10% smaller than the mean for the random approach. Between the two figures, this trial indicates that MC and MMC perform quite well compared to random in these mixed environments where m values differ and not all spectrum is common.

Figure 4.20 clearly indicates that the performance of MC and MMC is faster than random. The 25th, 50th, 75th, and 100th percentile values of the modular algorithms are lower than the random percentile values.

68
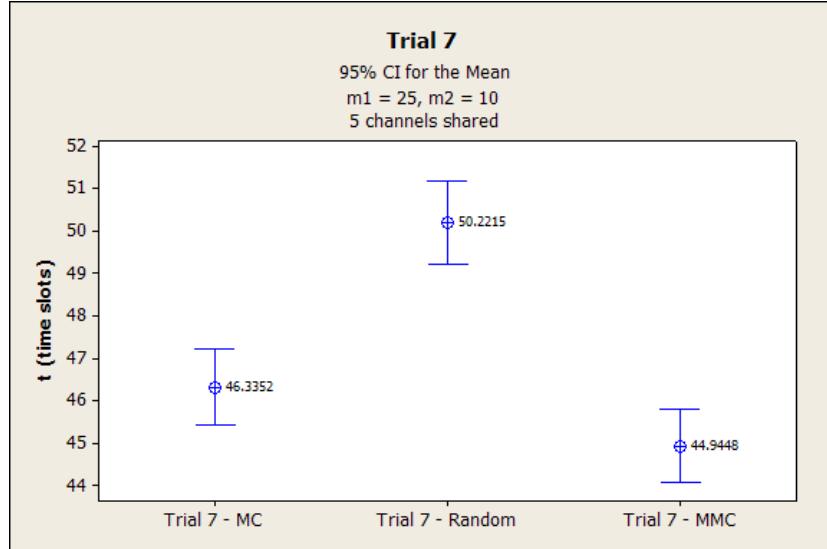
Figure 4.19: 95% Confidence Interval for $m_1 = 25$, $m_2 = 10$, $\bar{C} = 5$
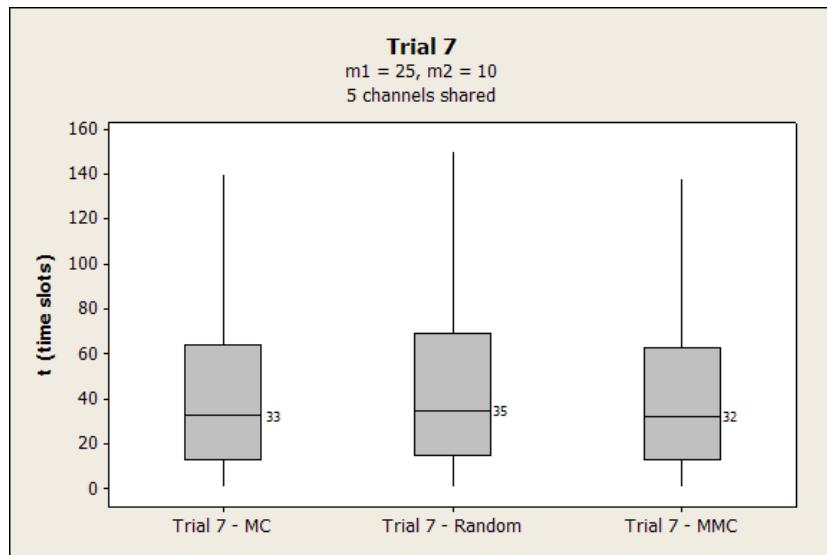
Figure 4.20: Box plot for $m_1 = 25$, $m_2 = 10$, $\bar{C} = 5$

Trial 8.   Trial 8 provides the final examination of the scenario discussed in Hypothesis 4. In this trial, $m_1 = m_2 = 25$ and $\bar{C} = 1$. Mathematically, the expected value of the trial would be $\frac{25 \cdot 25}{1}$, or 625.

Figure 4.21 only tells us that the mean value of MC is statistically greater than the mean values for random and MMC. Within 95% confidence, there is a very large overlap between the mean confidence interval of the random approach and MMC.

69
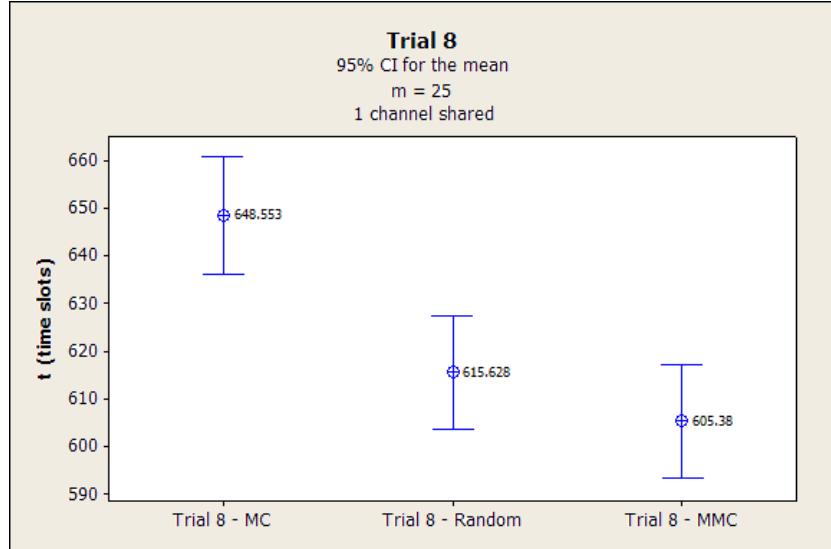
Figure 4.21: 95% Confidence Interval for $m_1 = 25$, $m_2 = 25$, $\bar{C} = 1$
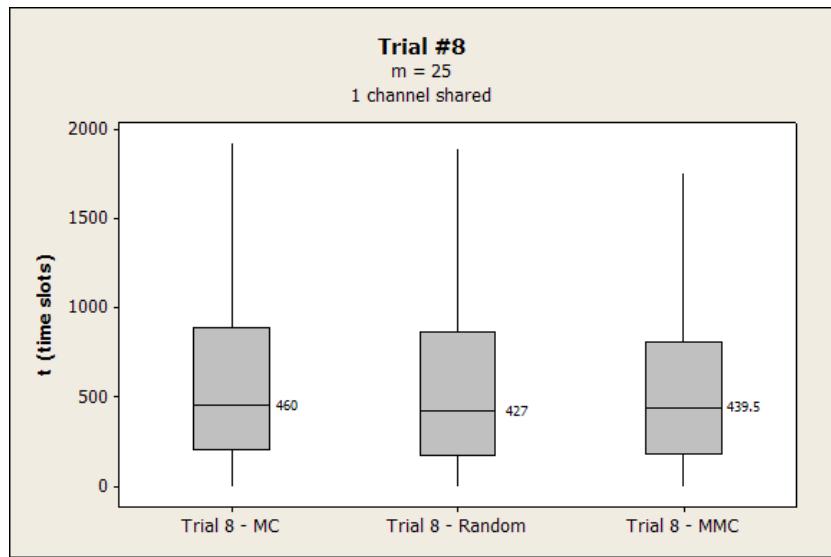
Figure 4.22: Box plot for $m_1 = 25$, $m_2 = 25$, $\bar{C} = 1$

Figure 4.22 provides some unexpected information. The median value of the random approach is smaller than the median value of the MMC, but the 75th percentile and 100th percentile values are larger for random than for MMC. Since the random approach lacks the guarantees of the CRT, this is somewhat to be expected. However, it is unexpected that random would perform better more than 50% of the time.

Hypothesis Results After Experiments

After analyzing the results of the 8 trials, we now revisit the hypotheses created before the experiments were conducted.

1. When $m_1 = m_2 < 9$ and $C_i = C_j$, random performs better than both MC and MMC. When $m_1 = m_2 > 9$ and $C_i = C_j$, MC will perform better than random as m increases.

   Result: confirmed. Trial 1 demonstrates that random had a lower mean TTR for $m < 9$, MC had a lower mean TTR for $m > 10$, and the crossover point is in the neighborhood of $m = 9$. As m values got larger in Trials 2 and 3, MC performed as much as 33% better than the random approach.

2. For $m_1 > 9$ and $m_2 > 9$, if the prime value selected under MC by radio 1 is equal to the prime value selected by radio 2 ($p_1 = p_2$) and neither $C_1 \subseteq C_2$ nor $C_2 \subseteq C_1$, then MMC performs better than MC.

   Result: confirmed. Trial 6 and Trial 8 demonstrate the MMC algorithm's superior performance when MC is likely to be in its degenerative case.

3. When there exists a large difference between m_i and m_j, random should perform better than MC and MMC.

   Result: refuted. Under Trials 5 and 7, there was a large difference between m_1 and m_2. However, in both trials MMC performed better than random, and MC performed as well or better. This tells us that the timeout function implemented by the algorithm is not likely to cause either algorithm to perform badly.

4. When $m_1 > 9$, $m_2 > 9$ and $m_1 \approx m_2$, as $\bar{C}$ gets smaller, MMC should perform better than random and MC.

   Result: refuted. The results from Trials 3, 4, 6, and 8 provide us with mixed results regarding this hypothesis. In some cases, the percentile values for MMC were lower than the other two. However, at no point in any of those three trials did MMC demonstrate that it clearly performed better.

V. Conclusions

This chapter summarizes the research presented in this thesis. It re-states the goals that went into developing the algorithms, and presents the results from the algorithm analysis and simulation. Finally, some ideas for follow-on research are proposed.

Research Goal

The goal of this thesis was to provide an efficient means to achieve spectrum rendezvous under hostile system models. The solution would analyze current approaches to rendezvous and provide a solution which improved upon the expected and guaranteed TTR under those models.

Research Conclusions

In very stable environments with a high percentage of common spectrum, the modular clock algorithm offers a 50% improvement in expected TTR over random channel visitation. In less stable environments, the modified modular clock algorithm provides the same expected TTR as random channel visitation while also guaranteeing rendezvous in $\theta(m^2)$ time with very high probability. These algorithms could be implemented using a layered approach, whereby the radio uses the modular clock algorithm earlier in its attempt to achieve rendezvous, then later using the modified modular clock or random visitation in the event that the modular clock algorithm fails to achieve rendezvous by a certain time.

Using game theory to analyze the Blind Rendezvous problem is very difficult due to the inability to observe the actions of the other players. However, game theory is very useful in the development of pursuit-evasion models and strategies to avoid jamming or other malicious activity.

72

Research Significance

This thesis presents the first wide reaching summary of rendezvous problems and solutions from a number of different contexts. The algorithms developed in this thesis are designed to perform well in environments that would be problematic for previously developed rendezvous algorithms. This thesis also formally defined many common problem models from cognitive network literature. By understanding the variables that significantly affect the problem models, more optimal solutions can be created for difficult problem models.

Future Research

The modified modular clock algorithm is one method to avoid and handle degenerative cases in the original modular clock implementation, but the large gap between the real number of channels and the prime number used causes the algorithm to choose randomly for a very large portion of the time. Further research into the optimal amount of prime numbers that should be available to the modified modular clock algorithm could improve performance.

Furthermore, more research should be done to apply concepts from designing cryptographic systems to the rendezvous problem. The problems of cryptography and rendezvous are not all that dissimilar. The modular clock algorithm leverages fundamental number theory to provide mathematical guarantees, much in the same manner cryptography relies on the number theory to provide guarantees about who is able to decode the traffic. Cryptography can also be handled through a variety of asymmetric or symmetric means, depending on the problem formulation.

Game theory is a good approach to observe the interactions between decision makers, but under the modular clock model there is little information in the spectrum for the radios to leverage to make decisions. The symmetric rendezvous-evasion game in this thesis gave an optimal mixed strategy for the pursuing player given the evading players' mixed strategy. However, the game did not consider the evading players'

optimal mixed strategy given the pursuing player's mixed strategy. These strategies should be analyzed until a Nash Equilibrium mixed strategy is found that is optimal for both sets of players.

Appendix A.   Simulation Source Code

This appendix provides the code used in generating the simulation data presented in chapter 4.

Simple Radio Class

Listing A.1:

```
1 #include <stdio.h>

  class Radio {
      public:

6         int rate;
          int m;
          int p;
          int position;
          int offset;
11        int startChannel;
          int endChannel;
          int timeStep    ;

          Radio(int, int, int);
16
          int advance();
  };
```

Listing A.2:

```
  #include "radio.h"
2 #include <math.h>

  Radio::Radio(int chanStart, int numChannels, int startPos) {
          startChannel = chanStart;
```

```
           m = numChannels;
7          position = startPos;
           offset = position - chanStart;
           endChannel = startChannel + m - 1;
           if (position > endChannel) {
                   printf("Channel bound error.  Setting position to ...
                        %i\n",endChannel);
12                 position = endChannel;
           }
           p = 0;
           rate = 0;
           timeStep = 0;
17 }


   int Radio::advance() {
           offset = fmod((offset+rate),p);
           position = offset + startChannel;
22         timeStep++;
           return(position);
   }
```

Random Rendezvous

Listing A.3:

```
1 #include <stdio.h>
  #include <stdlib.h>
  #include <math.h>
  #include <time.h>
  #include "radio.h"
6
  unsigned int getSeed() {
```

```
            FILE *file = fopen("/dev/random", "r");
            unsigned int temp;
            fread(&temp,4,1,file);
11          fclose(file);
            return(temp);
    }


    int gcd(int a,int b) {
16          int c;
            while(1) {
                    c = a%b;
                    if (c == 0) return b;
                    a = b;
21                  b = c;
            }
    }


    bool primeCheck(int x) {
26          for (int i = 2; i <= int(sqrt(x)); i++) {
                    if (gcd(i, x) > 1) return false;
            }
            return true;
    }
31
    int primeFind(int x, int bound) {
            while (x < bound) {
                    if (primeCheck(x)) return x;
                    x++;
36          }
            return 0;
    }


    int getArg(char *message) {
41          char inString[80];
            printf("%s",message);
```

```
            fgets(inString, 79, stdin);
            return(atoi(inString));
    }
46
    int main() {

            int startChannel, numChannel, position;

51          startChannel = getArg("Enter radio 1 channel start range\n...
                ");
            numChannel = getArg("Enter radio 1 number of channels\n");
            position = getArg("Enter radio 1 starting position\n");
            Radio radio1(startChannel, numChannel, position);

56          startChannel = getArg("Enter radio 2 channel start range\n...
                ");
            numChannel = getArg("Enter radio 2 number of channels\n");
            position = getArg("Enter radio 2 starting position\n");
            Radio radio2(startChannel, numChannel, position);

61          srand(getSeed());
            int i = 0;
            while(1) {
                    printf("At time %i, Radio 1 is at %i, Radio 2 is ...
                        at %i\n",i, radio1.position, radio2.position);
                    if (radio1.position == radio2.position) { printf("...
                        Rendezvous at t = %i\n on channel %i and %i\n",...
                        i,radio1.position, radio2.position); exit(0); }
66                  radio1.position = (rand() % radio1.m) + radio1....
                        startChannel;
                    radio2.position = (rand() % radio2.m) + radio2....
                        startChannel;
                    i++;
            }
    }
```

Modular Clock Rendezvous

Listing A.4:

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "radio.h"


unsigned int getSeed() {
        FILE *file = fopen("/dev/random", "r");
        unsigned int temp;
        fread(&temp,4,1,file);
        fclose(file);
        return(temp);
}


int gcd(int a,int b) {
        int c;
        while(1) {
                c = a%b;
                if (c == 0) return b;
                a = b;
                b = c;
        }
}


bool primeCheck(int x) {
        for (int i = 2; i <= int(sqrt(x)); i++) {
                if (gcd(i, x) > 1) return false;
        }
```

```
                return true;
30 }


   int primeFind(int x, int bound) {
           while (x < bound) {
                   if (primeCheck(x)) return x;
35                 x++;
           }
           return 0;
   }


40 int getArg(char *message) {
           char inString[80];
           printf("%s",message);
           fgets(inString, 79, stdin);
           return(atoi(inString));
45 }


   int main() {

           int startChannel, numChannel, position;
50
           startChannel = getArg("Enter radio 1 channel start range\n...
               ");
           numChannel = getArg("Enter radio 1 number of channels\n");
           position = getArg("Enter radio 1 starting position\n");
           Radio radio1(startChannel, numChannel, position);
55
           startChannel = getArg("Enter radio 2 channel start range\n...
               ");
           numChannel = getArg("Enter radio 2 number of channels\n");
           position = getArg("Enter radio 2 starting position\n");
           Radio radio2(startChannel, numChannel, position);
60
           srand(getSeed());
```

```
            radio1.rate = (rand() % (radio1.m - 2)) + 2;
            radio2.rate = (rand() % (radio2.m - 2)) + 2;    // Rates ...
                need to be between 2 and m-1


65          radio1.p = 0; radio2.p = 0;
            while (radio1.p == 0) {
                    radio1.p = primeFind(radio1.m,radio1.m*2);
            }
            while (radio2.p == 0) {
70                  radio2.p = primeFind(radio2.m,radio2.m*2);
            }

            printf("R1: %i, P1: %i, R2: %i, P2: %i\n",radio1.rate, ...
                radio1.p, radio2.rate, radio2.p);
            int i = 0;
75          while(1) {
                    printf("At time %i, Radio 1 is at %i, Radio 2 is ...
                        at %i\n",i, radio1.position, radio2.position);
                    if (radio1.position == radio2.position) { printf("...
                        Rendezvous at t = %i\n on channel %i and %i\n",...
                        i,radio1.position, radio2.position); exit(0); }
                    if (radio1.timeStep > 2*radio1.p) {
                            radio1.timeStep = 0;
80                          radio1.p = 0;
                            while (radio1.p == 0) {
                                    radio1.p = primeFind((rand() % ...
                                        radio1.m) + radio1.m, radio1.m...
                                        *2);
                            }
                            radio1.rate = (rand() % (radio1.m - 2)) + ...
                                2;
85                          printf("Radio 1 timed out.  New p: %i and ...
                                r %i\n",radio1.p, radio1.rate);
                    }
                    if (radio2.timeStep > 2*radio2.p) {
```

```
                        radio2.timeStep = 0;
                        radio2.p = 0;
90                      while (radio2.p == 0) {
                                radio2.p = primeFind((rand() % ...
                                    radio2.m) + radio2.m, radio2.m...
                                    *2);
                        }
                        radio2.rate = (rand() % (radio2.m - 2)) + ...
                            2;
                        printf("Radio 2 timed out.  New p: %i and ...
                            r %i\n",radio2.p, radio2.rate);
95              }
            radio1.advance();
            radio2.advance();
            if (radio1.position > radio1.endChannel) {
                    printf("First radio between real and prime...
                         at %i\n", radio1.position);
100                 srand(getSeed());
                    radio1.position = (rand() % radio1.m) + ...
                        radio1.startChannel;
            }
            if (radio2.position > radio2.endChannel) {
                    printf("Second radio between real and ...
                        prime at %i\n",radio2.position);
105                 srand(getSeed());
                    radio2.position = (rand() % radio2.m) + ...
                        radio2.startChannel;

            }
        i++;
110     }
    }
```

Modified Modular Clock

Listing A.5:

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "radio.h"


unsigned int getSeed() {
        FILE *file = fopen("/dev/random", "r");
        unsigned int temp;
        fread(&temp,4,1,file);
        fclose(file);
        return(temp);
}

int gcd(int a,int b) {
        int c;
        while(1) {
                c = a%b;
                if (c == 0) return b;
                a = b;
                b = c;
        }
}

bool primeCheck(int x) {
        for (int i = 2; i <= int(sqrt(x)); i++) {
                if (gcd(i, x) > 1) return false;
        }
        return true;
}
```

```c
    int primeFind(int m) {
            int i = 0;
            int primeArray[m];
            for (int j = m; j < 2*m; j++) {
                    if (primeCheck(j)) {
                            primeArray[i] =  j;
                            printf("Array %i is %i\n",i, j);
                            i++;
                    }
            }
            srand(getSeed());
            int randPrime = rand() % i;
            printf("Chose element %i, or %i\n",randPrime,primeArray[...
                randPrime]);
            return primeArray[randPrime];
    }


    int getArg(char *message) {
            char inString[80];
            printf("%s",message);
            fgets(inString, 79, stdin);
            return(atoi(inString));
    }

    int main() {

            int startChannel, numChannel, position;

            startChannel = getArg("Enter radio 1 channel start range\n...
                ");
            numChannel = getArg("Enter radio 1 number of channels\n");
            position = getArg("Enter radio 1 starting position\n");
            Radio radio1(startChannel, numChannel, position);
```

```
64          startChannel = getArg("Enter radio 2 channel start range\n...
                ");
            numChannel = getArg("Enter radio 2 number of channels\n");
            position = getArg("Enter radio 2 starting position\n");
            Radio radio2(startChannel, numChannel, position);


69          srand(getSeed());
            radio1.rate = (rand() % (radio1.m - 2)) + 2;
            radio2.rate = (rand() % (radio2.m - 2)) + 2;    // Rates ...
                need to be between 2 and m-1


            radio1.p = 0; radio2.p = 0;
74          while (radio1.p == 0) {
                    radio1.p = primeFind(radio1.m);
            }
            while (radio2.p == 0) {
                    radio2.p = primeFind(radio2.m);
79          }


            printf("R1: %i, P1: %i, R2: %i, P2: %i\n",radio1.rate, ...
                radio1.p, radio2.rate, radio2.p);
            int i = 0;
            while(1) {
84                  printf("At time %i, Radio 1 is at %i, Radio 2 is ...
                        at %i\n",i, radio1.position, radio2.position);
                    if (radio1.position == radio2.position) { printf("...
                        Rendezvous at t = %i\n on channel %i and %i\n",...
                        i,radio1.position, radio2.position); exit(0); }
                    if (radio1.timeStep > pow(radio1.p,2)) {
                            radio1.timeStep = 0;
                            radio1.p = 0;
89                          while (radio1.p == 0) {
                                    radio1.p = primeFind(radio1.m);
                            }
```

```c
                        radio1.rate = (rand() % (radio1.m - 2)) + ...
                            2;
                        printf("Radio 1 timed out.  New p: %i and ...
                            r %i\n",radio1.p, radio1.rate);
                }
                if (radio2.timeStep > pow(radio2.p,2)) {
                        radio2.timeStep = 0;
                        radio2.p = 0;
                        while (radio2.p == 0) {
                                radio2.p = primeFind(radio2.m);
                        }
                        radio2.rate = (rand() % (radio2.m - 2)) + ...
                            2;
                        printf("Radio 2 timed out.  New p: %i and ...
                            r %i\n",radio2.p, radio2.rate);
                }
                radio1.advance();
                radio2.advance();
                if (radio1.position > radio1.endChannel) {
                        printf("First radio between real and prime...
                             at %i\n", radio1.position);
                        srand(getSeed());
                        radio1.position = (rand() % radio1.m) + ...
                            radio1.startChannel;
                }
                if (radio2.position > radio2.endChannel) {
                        printf("Second radio between real and ...
                            prime at %i\n",radio2.position);
                        srand(getSeed());
                        radio2.position = (rand() % radio2.m) + ...
                            radio2.startChannel;


            }
        i++;
        }
```

```
119 }
```

Bibliography

1. S. Alpern and S. Gal. *The Theory of Search Game and Rendezvous.* Springer, 2003.

2. E.J. Anderson and R.R. Weber. The rendezvous problem on discrete locations. *Journal of Applied Probability and Statistics*, 28:839–851, 1990.

3. K. Balachandran and J.H. Kang. Neighbor discovery with dynamic spectrum access in adhoc networks. *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, 2:512–517, May 2006.

4. Milind M. Buddhikot, Paul Kolodzy, Scott Miller, Kevin Ryan, and Jason Evans. Dimsumnet: New directions in wireless networking using coordinated dynamic spectrum access. In *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pages 78–85, Washington, DC, USA, 2005. IEEE Computer Society.

5. Lili Cao and Haitao Zheng. Distributed spectrum allocation via local bargaining. *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, pages 475–486, 26-29 Sept., 2005.

6. Luis DaSilva. Rendezvous in cognitive radio networks, 2008.

7. Luis DaSilva and Igor Guerreiro. Sequence-based rendezvous for dynamic spectrum access. *IEEE DySpan 2008*, 2008.

8. Bruce Fette. *Cognitive Radio Technology.* Newnes, 2006.

9. B. Horine and D. Turgut. Link rendezvous protocol for cognitive radio networks. *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 444–447, April 2007.

10. Pil Jung Jeong and Myungsik Yoo. Resource-aware rendezvous algorithm for cognitive radio networks. *International Conference on Advanced Computing Technologies*, February 2007.

11. D. Kowalski and A. Malinowski. How to meet in an anonymous network. *Theoretical Computer Science*, 2008.

12. Wei Shi Lim. A rendezvous-evasion game on discrete locations with joint randomization. *Advanced Applied Probability*, 29:1004–1017, 1997.

13. Allen B. MacKenzie and Luiz A. DaSilva. *Game Theory for Wireless Engineers.* Morgan Claypool, 2006.

14. Richard Martin. Rendezvous sequences. Written Correspondence, June 2008.

15. N. Nie and C. Comaniciu. Adaptive channel allocation spectrum etiquette for cognitive radio networks. *New Frontiers in Dynamic Spectrum Access Networks,*

2005. DySPAN 2005. 2005 First IEEE International Symposium on, pages 269–278, 8-11 Nov. 2005.

16. A. B. Mackenzie R.W. Thomas, L. A. DaSilva. Cognitive networks. *IEEE DySPAN*, pages 352–360, 2005.

17. Bluetooth SIG. How bluetooth works - communications, 2008. http://www.bluetooth.com/Bluetooth/Technology/Works/Communications_Topology.htm.

18. Mark D. Silvius, Feng Ge, Alex Young, Allen B. MacKenzie, and Charles W. Bostian. Smart radio: spectrum access for first responders. *Wireless Sensing and Processing III*, 6980:698008, 2008.

19. Mark D. Silvius, Allen B. MacKenzie, and Charles W. Bostian. A survey of neighbor discovery protocols in modern wireless systems and their application to the design and implementation of "rendezvous" algorithms in smart and cognitive radios. *Virginia Tech Symposium on Wireless Personal Communications*, 2007.

20. V. Srivastava, J. Neel, A.B. Mackenzie, R. Menon, L.A. Dasilva, J.E. Hicks, J.H. Reed, and R.P. Gilles. Using game theory to analyze wireless ad hoc networks. *Communications Surveys and Tutorials, IEEE*, 7(4):46–56, Fourth Quarter 2005.

21. P.D. Sutton, K.E. Nolan, and L.E. Doyle. Cyclostationary signatures for rendezvous in ofdm-based dynamic spectrum access networks. *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 220–231, 17-20 April 2007.

22. David Wells. *The Penguin Book of Curious and Interesting Numbers: Revised Edition*. Penguin Press Science, 1986.

Vita

Captain Nicholas C. Theis graduated from West Ottawa High School in Holland, Michigan. He entered undergraduate studies at Grand Valley State University, completing a minor in mathematics before transferring to the University of Michigan. At Michigan, he graduated with a Bachelor of Science degree in Computer Science in May 2002. He was commissioned through Officer Training School in June 2004 as part of class 04-06. His first assignment was at Randolph AFB as a Crew Commander in the Air Education and Training Command Network Operation and Security Center. Before leaving Randolph, Captain Theis was honored as the Headquarters Air Education and Training Command Directorate of Communications Company Grade Officer of the Year for 2006. In September 2007, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the 57th Information Aggressor Squadron at Nellis AFB.

Permanent address: 2950 Hobson Way
                         Air Force Institute of Technology
                         Wright-Patterson AFB, OH 45433

1. REPORT DATE *(DD-MM-YYYY)* 27-02-2009	2. REPORT TYPE **Master's Thesis**	3. DATES COVERED *(From – To)* September 2007 – March 2009

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Modular Clock Algorithm for Blind Rendezvous	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER JON 09-265
Theis, Nicholas C., Capt, USAF	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765 DSN: 785-3636	AFIT/GCS/ENG/09-08

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RYRE
Air Force Research Laboratories Attn: Vasu Chakravarthy Sensors Directorate 2241 Avionic Circle WPAFB OH 45433 DSN: 785-5579 x 4245 vasu.chakravarthy@wpafb.af.mil	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A new algorithm, the modular clock algorithm, is developed and analyzed as a solution for the simple rendezvous environment model, coupled with a modified version for environment models with less information. This thesis examines the rendezvous problem as it exists in a Dynamic Spectrum Access cognitive network. Specifically, it addresses the problem of rendezvous in an infrastructureless environment. The thesis includes a taxonomy of commonly used environment models, and analysis of previous efforts to solve the rendezvous problem. Mathematical models and solutions used in applied statistics are analyzed for use in cognitive networking. A symmetric rendezvous pursuit-evasion game is developed and analyzed. Analysis and simulation results show that the modular clock algorithm performs better than random under a simple rendezvous environment model, while a modified version of the modular clock algorithm performs better than random in more difficult environment models.

15. SUBJECT TERMS
Cognitive radio, cognitive network, dynamic spectrum access, rendezvous

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON **Capt Ryan W. Thomas, PhD (ENG)**
REPORT **U**	ABSTRACT **U**	c. THIS PAGE **U**	UU	103	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636; email: ryan.thomas@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18